

ADAT- ÉS PROGRAMVÉDELEM IBM PC-RE

LSI ALKALMAZÁSTECHNIKAI
TANÁCSADÓ SZOLGÁLAT



VAGYONVÉDELEM

Védje meg!

Kapuk,
portálok,
kirakatok,
védőrácsok,
detektívrácsok,
betörésbiztos ajtók

berendezéseit,
számítógépeit,
értékeit

készítését vállaljuk

Érdeklődni lehet: 805-713, 889-751
Budapest III. Hévízi u. 6/e.

Levélcím: 1300 Budapest,
Postafiók 114.

Ráth György

ADAT- ÉS PROGRAMVÉDELEM IBM PC-RE

LEKTORÁLTA

Éltető László

ISI

ALKALMAZÁSTECHNIKAI TANÁCSADÓ SZOLGÁLAT
BUDAPEST, 1989

Kiadó: LSI Alkalmazástechnikai Tanácsadó Szolgálat
Felelős kiadó: dr. Kovács Magda
Témafelelős: Székely László
Isbn: 963 592 902 1
Eng. szám: 16 275
Készült: BÁSTYA—Textprint
Felelős vezető: Balanyi István

Tartalomjegyzék

Bevezetés	4
Mikrogépes adat- és programvédelem	4
I. Az IBM PC/XT felépítése, memóriatérképe	10
Az IBM PC/XT interrupt vektorai	12
Rövid bevezetés a floppy lemezekhez	13
A DOS lemezek formátuma	18
II. A másolóprogramok és másolást segítő eszközök rövid áttekintése	21
Hogyan működik a másolás elleni védelem?	22
Másolásvédelmi rendszerek	24
PROLOK	24
SUPERLOCK	35
III. ENCRYPTION védelem	45
IV. Adatvédelmi rendszerek	49
LAN	49
LE MENU	50
SAFE-GUARD	51
EltSECRET	57

Bevezetés

Mikrogépes adat- és programvédelem

A mikrogépes adat- és programvédelem története és fejlődése szorosan összefügg a mikrogépek fejlődésével, sokszor vált döntővé egy-egy új mikrogép megjelenése a piacon, és messzemenően megváltoztatta még a tőle jelentősen eltérő gépekhez kapcsolódó védelmi stratégiákat is.

A legelső mikrogépek (pl. ALTAIR) alig hasonlítottak a maiakhoz: olyan jellemző részegységek hiányoztak belőlük, mint a billentyűzet, a mágneslemez-meghajtók, stb. Nem kellett azonban sokáig várni, s 1977. körül megjelent az első, sorozatban gyártott mikrogép, az Apple-II. Kitört a "mikroláz". Az egyre terebélyesedő Apple mellett megjelent sok más kisebb-nagyobb cég. A mikropiac teljesen kaotikus volt, de több tendenciát vehettünk észre:

- megkezdődött a számítógépek szétválása játék, hobby és üzleti célú gépekre;
- a processzorok között bizonyos szelektálódás játszódott le, néhány év alatt már csak alig féltucat gyártótól szereztek be a számítógépes cégek a "gép lelkét";
- a jó konstrukció csak fél sikert jelentett. Nem volt mindegy, hogy milyen szoftver-támogatást kaptak a vásárlók. Ha a gyártó cég üzletpolitikája sikeres volt, egyre többen vettek meg gépet. Ez a "külső" programfejlesztőket is arra serkentette, hogy egyre több programterméket fejlesszenek ki az illető géptípusra. Ez viszont tovább növelte a gép "túlélési" esélyét a számítógépes piacon.

Így történhetett meg, hogy pár év alatt az amerikai piacon uralkodóvá váltak az Apple-II, továbbá a TRS-80 gépekre írt szoftverek, valamint olyan programtermékek, melyek a DIGITAL RESEARCH CP/M operációs rendszere alatt futottak. Utóbbiak érdekessége, hogy míg az Apple-II és a TRS-80 számítógépeket csak a névadó két cég gyártja (a távolkeleti másolóktól eltekintve – ld. később), a CP/M operációs rendszer kisebb-nagyobb változatai több tucat gyártó teljesen különböző gépein futottak (ezeknek a gépeknek sokszor csak a mikroprocesszora volt azonos, kezdetben az INTEL 8080-as processzor, később a Z80-as).

Kezdetben a mikrogépes védelem kérdése fel sem merült: a "hőskorszakban" rendkívül szoros volt az együttműködés a számítógép-fejlesztők, programfejlesztők és a felhasználók között. Kevés mikrogépet dobtak a piacra, és a kereskedelemben forgalmazott programok köre is szűk volt. Az adatvédelem nem is létezett még mikrogépekre: a viszonylag csekély kapacitású floppylemezek (kezdetben alig 100K!), a nehézkesen kezelhető magnókazetták nem számítottak komoly adattárolóknak.

De mindez hamarosan megváltozott: az Apple-gépeket egyre többen vásárolták, már számítógépes játékokra (sőt, sokszor csak arra!) használták őket, és a játékokat kezdték a felhasználók másolni, egymás között csereberélni. És itt már nem volt megállás: hiszen egy egyszerű számítógépes játék lemásolása semmiben sem különbözik egy komoly felhasználói rendszer lemásolásától. Az utóbbiak viszont már nem filléres áruk: érdemes tehát őket pénzért továbbadni! Miután pedig egyszerű mikrogéppel rá lehetett kapcsolódni a tele-

fonvonalakat felhasználó különböző nagygépes rendszerekre, sokan nagy zavart idéztek elő a nagyszámítógépes adatbankokban. Így azután elkezdődött a mikroszámítógépek világában is a védő és a védelmet hatástalanító rendszerek egymást erősítő versenye. Itt hangsúlyozzuk azt az egyébként triviális tényt, hogy egy mikrogép-családra alkalmazott védelmek bonyolultsága és népszerűsége elsősorban nem a gépek árával, ill. a programok árával van arányban, hanem azok elterjedtségével: ezt fényesen bizonyítja az, hogy az akkor igen drága CP/M-es gépekre nem fejlesztettek ki igazán említésre méltó védelmet. (Ugyan számos gép használta a CP/M-et, s a rendszer is népszerű volt, de a legtöbb gép még a legalapvetőbb paramétereiben sem volt a többi géptípussal kompatibilis: géptípusonként más volt a floppy mérete, a floppy felírás formátuma, a képernyőméret, a billentyűzet. És bár a CP/M volt a "közös nevező", a programokat nem lehetett egyszerűen egyik gépről a másikra áttenni. Ennek a világnak az utolsó mohikánjai itt-ott Magyarországon is fellelhetők: az OSBORNE, KAYPRO-II, TRANSMIC-8, PROPER-8, MOD-81, TERTA, M08X mind megannyi különböző, jobb-rosszabb konstrukció, voltak egészen sikeresek is köztük: de csak a CP/M kapcsolta őket össze, igen lazán.)

Ezzel szemben az Apple-II világában a programvédelmek és az azokat kijátszó módszerek rendkívül dinamikusán fejlődtek, és pár év alatt rendkívül magas szintre jutottak el.

A mikrogépes adatvédelmi eljárások azonban csak a gyorsabb, nagyobb kapacitású és viszonylag olcsó tárolók (winchester, azaz merevlemez) megjelenésével kezdtek terjedni.

A programvédelem teljesen "skizofrén" példája az olcsó játékgépekre írt programok védelme: igen ravasz eljárásokkal próbálják elejét venni annak, hogy a pár dolláros játékprogramok illegálisan terjedjenek. Itt megemlítendőek a mágneskazettás (!) védelmek ZX Spectrumra, amikor is mindent megtesznek annak érdekében, hogy a játékot ne lehessen a memóriából kimenteni, másolóprogrammal másolni, s így az egyszerű felhasználónak nem marad más hátra, mint a szalagot magnóról magnóra másolni, ami viszont minőségromláshoz vezet. (Főleg akkor, ha a programszalag nagyfrekvenciás jelekkel van tele!) A szintén "népgépnek" számító Commodore-64 floppys védelmei virtuozitásukban szinte felülmúlhatatlanok: szinte hihetetlen, hogy egy-egy, sokszor gyengébb játékprogram kedvéért mi mindent el nem követnek a gyártók, hogy a programot ne lehessen másolni, s a felhasználók, hogy a védelmet "feltörjék".

A nagyfrekvenciás módszerrel dolgozó, 1984-ben megjelent angol SPEEDLOCK védelmi eljárás kifejlesztése (ZX SPECTRUM gépre!) hozzávetőleg annyi munkát igényelhetett, mint egy akkori IBM PC-s fejlesztés, a PROLOK. A különbség "csupán" az, hogy az előbbi egy mókás foci-programot (MATCH DAY) védett, melynek ára néhány hét után kb. 5 angol font volt, az utóbbi pedig a dBASE III programcsomagot védte, melynek ára ma is 400 és 700 USD között (tehát 250-500 angol font) mozog. (Sokszor a védelmek odáig "fejlődtek", hogy a védelem zavarta a szoftver rendeltetésszerű használatát is sajnos, a védelmeknek ez a túlhajtása időnként a fejlettebb gépekre írt szoftverek esetén is jelentkezik /ld. később/.) Természetesen a játékgépekre is igaz a "népszerűségi" törvény: míg a SPECTRUM és COMMODORE gépek programjainak védelmével, illetve azok feltörésével a világ szinte minden országában foglalkoztak, addig ki emlékszik már az egyébként nem is olyan rossz DRAGON-32 illetve DRAGON-64 gépekre írt védelmekre? Ha egyáltalán létezett ilyen védelem.

Az Apple-II sikere nem csak a hobby- és az üzleti, hanem a játékgépek frontján is meggyőző volt: az USA-ban a nyolcvanas évek elején a személyi számítógép piacnak több, mint felét uralta! Ehhez hozzájárult egy addig nem tapasztalt jelenség: megjelentek a piacon a különböző ORANGE, APOLLO, PINEAPPLE stb. fantázianevű gépek, melyek árfekvése a játékgépekével egyezett, viszont reklámjuk azt hirdette, hogy teljesen kompatibilisek az APPLE-II-vel. Ezek a gépek távol-keleten (Japán, Tajvan, Hong-kong, Dél-Korea, Szingapur) készültek, és — legalábbis az árukhoz képest — meglepően jók voltak. Az így több millióssá vált Apple-II piacon tehát újabb lendületet vett a védelem-feltörés verseny. A szerzői jogi problémákról nem is beszélve, a személyi számítógépek gyártói először néztek szembe ilyen gépdömpinggel. Ez a gépinvázio a csendes-óceáni térségben (és részben az USA-ban is) azt eredményezte, hogy ott sohasem jutottak olyan szerephez a hagyományos "játékgépek", mint Európában: a diákok is inkább megvettek ugyanazért a pénzért egy APOLLO-II hong-kongi gépet (és ebben az iskola is támogatta őket!), mint egy COMMODORE-64-et.

1981-ben a személyi számítógép gyártásba beszállt a tőkeerős amerikai óriáscég, az IBM. Egy nem túl jól felszerelt, ámde rendkívül drága gépet sikerült kihozniuk, melynek nem jósoltak a szakértők nagy jövőt. Két erős pontja volt ennek a gépnek: az első az akkor viszonylag jónak számító INTEL 8088-as mikroprocesszor, melynek címezési tartománya 1 Mbyte, a gép viszont alapkiépítésben csak 16Kbyte memóriával rendelkezett, s legfeljebb 64 K-ig volt bővíthető. A másik jó "húzást" az Apple-II-től lesték el: a gép öt úgynevezett slot-tal (rés, ahová bővítőkártyát lehet csatlakoztatni) rendelkezett, miáltal nyitva állt az út a "kívülálló" cégek előtt, hogy különböző perifériákat fejlesszenek ki erre a géptípusra.

A processzor kiválasztása abból a szempontból is szerencsés volt, hogy az azelőtt meglévő, INTEL 8080 alapú CP/M alatt futó programokat a fejlesztő cégek viszonylag könnyen átültethették az IBM gépére, amelyen ráadásul a CP/M-re sok tekintetben hasonlító PC-DOS operációs rendszer futott. (Ezt a rendszert a MICROSOFT cég fejlesztette ki.) Az IBM saját személyi számítógépét IBM-PC névre keresztelte. Ezt a gépet hamarosan követte a már 256K-ra bővíthető IBM-PC-II és a külső hardver-gyártók számos bővítőkártyája, akárcsak az Apple-II esetében. Nyilvánvalóvá vált, hogy az IBM-PC — legalábbis az üzleti életben, irodákban alkalmazott mikrogépek területén — komoly versenytársa az addig a piacot uraló CP/M-es gépeknek. Szemben az utóbbiak sokféleségével, az IBM-PC szabványt tudott kialakítani: az új IBM mikrogép típusok felülről kompatibilisek lettek a régiekkel. Ráadásul rövid idő alatt az összes, CP/M alatt futó programot (az IBM támogatásával) átültették IBM-PC-re. Elindult tehát az önmagát erősítő folyamat, az IBM hardver és a rajta futó szoftverek párhuzamos felfutása, az ismert koreográfia szerint. Hamarosan megjelent az első, nem IBM-től származó, de vele teljesen kompatibilis számítógép, a COMPAQ. Az IBM 1982. végén kihozta az alapmodell harmadik, javított változatát, ahol is az addigi öt slot helyett 7 slot, pontosabban 7 és fél slot volt (7 teljes értékű és egy speciális slot, rövidebb kártyák számára, melynek hardver paraméterei is eltérnek a többi slot-étól, így ritkábban használják a bővítőkártyák). Az új modellben jelentős változás volt az is, hogy az eddigi két floppy meghajtó helyett egy floppy meghajtóval és egy merevlemez tárolóval (winchester) rendelkezett. Ezt a modellt IBM-PC/XT névre keresztelték, s ez volt az a típus, mely közvetlen elődjével, az IBM-PC-II-vel meghódította később a világot.

Jól átgondolt lépés volt az, hogy nagy súlyt fektettek a kompatibilitásra: még a legújabb gépeken futó rendszerek is futtathatóak a (kellően kibővített) IBM-PC-I-eken, tehát az alapmodellen, s természetesen fordítva is: a régi szoftverek futnak az új gépeken. A COMPAQ gépet több amerikai cég IBM-kompatibilis gépe követte. Ezeken a MICROSOFT cég MS-DOS operációs rendszerei futottak, melyek ugyan bizonyos értelemben gép-specifikusak voltak, de a legtöbb program változtatás nélkül futott mindegyik gépen. Meg is indult a versengés a gyártók között: akinek a gépén több IBM-program futott változtatás nélkül, az "IBM-kompatibilisebbnek" kiáltotta ki termékét, s ez jó reklámfogásnak bizonyult. Logikus tehát, hogy a programvédelmek (és az őket kijátszó másoló és feltörő programok) soha nem látott virágzásnak indultak. A távol-keleti gyártók is rájöttek, hogy ha az Apple-kompatibilis gépek mintájára IBM-kompatibilis gépeket gyártanak, a már meglévő óriási szoftver-választék garancia arra, hogy a piac a gépeket kedvezően fogadja. Ennek következtében az IBM-PC/XT kompatibilis gépek újabb hulláma árasztotta el a piacot, s a gépárak zuhanni kezdtek. A távol-keleti PC/XT gépek árai nemcsak az Apple-II és a CP/M gépek, hanem a játékgépek áaira is rázuhantak: jeleneg egy kétfloppys, 640 Kbyte memóriakapacitású gép ára az USA-ban 500 \$ körül mozog, ugyanannyi az ára távol-keleten, ami kb. 800 DM-nak felel meg. Egy COMMODORE-64 gép ára 64(!) Kbyte memóriával és két meghajtóegységgel kb. $400 + 500 + 500 = 1400$ DM, tehát majdnem kétszerese egy tajvani PC/XT árának. És a két gép szolgáltatásait, sebességét, memóriakapacitását össze sem lehet hasonlítani.

Végül egy "övon aluli" ütés: hogy az IBM jogilag ne támadhassa meg a távol-keleti gépmásolókat, a távol-keleti PC/XT gépek csak olvasható memóriájába (ROM-jába) általában eléggé szedett-vedett program van gyárilag beégetve. Ha veszünk egy ilyen gépet, megkérdejelezhethetjük, hogy kompatibilis-e az IBM-mel. Azonban közkézen forog a recept: mihelyt a vásárló hazaviszi a gépét, a ROM chipet azonnal kidobja, vesz megfelelő számú új ROM chipet (melyek még a magyar kereskedelemben is kaphatóak, áruk 500 Ft körül van!), általában öt darabot, és házilag néhány perc alatt beégeti az eredeti IBM-féle ROM programot. Ezután nyugodtan úgy veheti, hogy eredeti IBM-gépe van (legalábbis az alkártya eredeti. A bővítő egységek gondos kiválasztásával az IBM-PC/XT-vel teljesen kompatibilis, sőt sok tekintetben annál jobb gép állítható össze). Így a nyolcvanas évek közepére — második felére egy több tíz milliós IBM felhasználói kör jött létre szinte a semmiből. Több ezer, ha nem tízezer üzleti és játékprogram jelent meg a piacon, magával hozván a védelmeket és az azokat kijátszó programokat.

Magyarországon a kezdeti próbálkozások után egyértelműen a COMMODORE-64 időszaka köszöntött be. Szerencsére azonban mára ez a játékgép az üzleti életből lassan kiszorul, sőt az igényesebb számítógép-kedvelők és programozók otthonában is egyre gyakrabban előfordul ha nem is IBM-PC/XT, de legalább azzal kompatibilis, olcsó távol-keleti számítógép.

Számítanunk kell tehát arra, hogy az évtized végére az IBM-kompatibilis gépek egyeduralkodóak lesznek Magyarországon is a közepes kategóriájú személyi számítógépek piacán. (Kérdés, hogy az olcsóbb számítógépek, s az iskolagépek kategóriáját a régebbi gépek meddig fogják uralni.) Így óhatatlanul a szerzői jogi kérdések, az adatvédelmi problémák és minden olyan probléma, mely a hozzáférés, birtoklás és használati jog körül kialakulhat, a jövőben éppen az IBM-kompatibilis gépek családjában fog tetőzni, ahogy jelenleg a fej-

lett országokban tapasztalható. Éppen ezért a mikrogépes adat- és programvédelem témakörében az IBM-kompatibilis gépekkel, köztük is a leginkább elterjedtekkel, az IBM-PC/XT-vel érdemes foglalkozni.

Tekintsük át a mikrogépes információvédelem (tehát program- és adatvédelem) jellemző vonásait.

A gépek többnyire személyi jellegűek, ami az információvédelmet a személyi tulajdon védelmének irányába befolyásolja. A számítógép jogtalan eltulajdonítása ellen számítógépes módszerekkel nehéz fellépni, de már a jogtalan használat ellen számtalan módszert találtak ki. Csak néhányat sorolnék itt fel:

- az IBM-PC/AT szabályosan "kulcsra zárható". Többféle megoldást árulnak a nyugati cégek az IBM-PC/XT-hez is. Közös jellemzőjük az áramforrás és a tápegység közé ékelődő zárszerkezet, mely csak akkor engedi a gépet működni, ha kulccsal kinyitják. A zárszerkezetet csak hosszadalmas szereléssel lehet a gépből eltávolítani. (Még tovább megy az IBM AT védelme: a zárszerkezet zárt állapotban nem engedi, hogy a billentyűzetet használják vagy a gépfedelet leemeljék.)
- több olyan segédprogram létezik, melyek megakadályozzák, hogy illetéktelen személy a merevlemezre, illetve a floppy lemezre írjon (pl. lightning).
- Ha a rendszer partícióinkat nem boot-olhatóként hoztuk létre (ezt meg tudjuk tenni!), csak a rendszerlemez birtokában lehet a rendszert elindítani.
- Ha egy rövid programmal a Winchester partíciótábláját rontjuk el (pl. minden byte-ját XOR-oljuk valami konstanssal), még rendszerlemez birtokában sem tudnak a Winchesterhez hozzáférni, a DOS "nem ismeri fel" a lemezt. Csak úgy tudjuk megint használhatóvá tenni, ha – floppyról bootolva – ismét lefuttatjuk programunkat, s az visszaállítja a partíciótábla eredeti alakját. Ezután ismét lehet a Winchesterről bootolni.

A szoftverek védelmére kidolgozott módszereket több szempontból csoportosíthatjuk, legkézenfekvőbb az információ-hordozó eszközök szerint osztályozni. Ennek megfelelően a következő eszközök védelmével foglalkozunk:

- floppy (hajlékony) lemez, mely mikrogépek esetén talán a legfontosabb, ugyanakkor a legkönnyebben védhető tároló.
- winchester (merevlemez), mely a floppy-nál helyhez kötöttebb, különleges védelmi módszereket igényel, s a tárolók közt a legnehezebben védhető.
- újabban nagyon sok szoftverhez adnak ún. "hardware-lock" berendezést, mely a nyomtató vagy az RS232 kimenetre dugható be. Két típusa létezik:
 - fix válaszó, amikor a program "rákérdez" a fekete doboz (hardwarelock) tartalmára;
 - algoritmusos, amikor a fekete doboz tartalmát a program dinamikusan használja fel működése során.

A hardwarelock előnye, hogy a program futás közben sokszor, több pontról ellenőrizheti, hogy megvan-e.

- kulcskártya, mely egy ROM segítségével a szoftver egy (kulcsszerepet játszó) részét tartalmazza. Az ilyen kártya előállítása bonyolultabb, másolásához speciális berendezésre van szükség.

A fentieken kívül természetesen más védelmi stratégiák is kialakultak. Ezek bizonyos esetekben — amire általában csak igen nagy szabadpiacon van lehetőség — a termék árával manipulálnak. Ha ugyanis ez elég alacsony, akkor olcsóbb megvenni, mint lemásolni a szoftvert.

Olyan esetekre is van példa, hogy nem is a programot, hanem a dokumentációját védik — egy speciális bevonattal másolás ellen. Ezzel a szoftver eltulajdonítása értelmetlenné válik, hiszen már viszonylag egyszerű rendszerek sem használhatóak felhasználói kézikönyv nélkül. Például a vörös alapszínen levő fekete betűk sem "jönnek ki" a fénymásolás során, így az ilyen dokumentációt nem lehet XEROX-ozni, azonban elolvasása is nagyon fárasztó a szemnek.

I. Az IBM PC/XT felépítése, memóriatérképe

Az IBM-PC/XT (vagy azzal kompatibilis számítógép) legfontosabb része egy 8.5 x 12 inch (21.6 x 30.5 cm) méretű áramköri kártya, az úgynevezett alaplapon. Az alaplapon négy funkcionális egység foglal helyet:

- Az Intel 8086-os processzor (több, kompatibilis gépnél Intel 8088-as processzor) és az azt közvetlenül kiszolgáló egyéb áramkörök.
- Intel 8087-es matematikai coprocesszor (ha van)
- Csak olvasható memória (ROM), mérete legfeljebb 64 K (Részletesen ld. később.)
- írható/olvasható memória, max. 640 K, ám némely modellben (PROPER-16) 704 K-ig terjed. Ez azonban inkább hátrány, mint előny, ugyanis újabban több képernyőadapter-típus (pl. EGA) is használja a 640K-704K közötti területet.
- I/O-t (ki/bemenetet) támogató áramkörök és egységek. Ezek közül a legszembetűnőbb a (korábbi PC verzió esetén 5), XT-nél 8 db úgynevezett I/O SLOT (rés), amelyekbe különböző bővítő kártyák helyezhetők be, éppen ebben rejlik a PC/XT rendszer rugalmassága.

A memória felépítése a következő:

0 K – 640 K:	írható/olvasható memória
704 – 720 K:	monokróm képernyő adapter memória
736 – 752 K:	színes képernyő adapter memória
800 – 808 K:	winchester meghajtó saját ROM-ja, egyes esetekben ez 800-816 K, illetve 800-804 K is lehet.
960 – 1024 K:	ROM

Megjegyzések:

- 1) A korábbi modelleken az alaplapon csak 256 K írható/olvasható memória (a továbbiakban: RAM) számára volt hely, újabb modelleken 640 K számára van hely.
- 2) Az Intel 8086 processzor 1 MB (1024 K) területet, azaz hexadecimálisan 0-FFFFF-et képes címezni, azonban 64 K-s ún. szegmensekben. Például a hexadecimálisan 4200:3140-ként ábrázolt cím valódi értékét a következőképpen kapjuk meg:

$$\begin{array}{r} 42000 \\ + 3140 \\ \hline 45140 \end{array}$$

Ebből a 4200-at szegmensnek, a 3140-et offset-nek nevezzük. Ez a címzési mód a 8088 regiszterei segítségével történik, melyek 16 bitesek és köztük a DS, ES, SS, CS a szegmensregiszterek, a többi pedig (AX, BX, CX, DX, SP, BP, SI, DI) az egymás

közötti műveletekre és az offset címzésre használatos. (Van ezenkívül egy FLAG regiszter, amely lényegében a processzor állapotregisztere.)

- 3) Látható, hogy a 640 K feletti címzési terület meglehetősen "lyukas". Ez azzal magyarázható, hogy különböző területek a későbbi fejlesztések számára vannak fenntartva. Néhány cég megpróbálkozott a 640 – 704 K közti memória terület kihasználásával (pl. az SZKI egyes PROPER modelljei), ez azonban az újabb képernyő-adaptereknél konfliktusokhoz vezet (pl. EGA grafika).
- 4) A ROM két, jól elkülönülő részre tagolható:
 - egy egyszerű BASIC interpreter
 - BIOS (a gép alapvető input/output/ működtetését végző modul); ez tartalmazza a programvédelmekben is használt funkciót, a floppy lemez kezelését. Megjegyezzük, hogy azok a védelmek, melyek a BIOS-t használják a lemez kezelésekor, általában a gyengébbek közé tartoznak.
- 5) A memóriafelépítés gyakrabban alkalmazott szegmens-offset ábrája:

0 : 0

RAM terület

9000 : FFFF

B000 : 0

monokróm képernyő

B000 : 3FFF

adapter memória

B000 : 8000

színes képernyő másként:

B800:0

B000 : BFFF

adapter memória

B800:3FFF

C000 : 8000

winchester meghajtó

C800:0

C000 : BFFF

saját ROM-ja másként:

C800:3FFF

F000 : 0

BASIC ROM és
BIOS ROM

F000 : FFFF

A BASIC és BIOS ROM pontos elhelyezkedése a következő: Basic ROM F600:0 és F600:7FFF között, a BIOS az F000:E000 és F000:FFFF között van. Ha utánaszámolunk, látjuk, hogy a BIOS rögtön a BASIC után következik. Felmerül azonban a kérdés, hogy mi is van F000:0 és F000:6000 között? A válasz meglepő: az eredeti PC/XT-ben két db. ROM IC számára van hely, melyek így szükségszerűen 256 Kbites chipek. $((256 \times 2) : 8 = 64)$ Mivel más megoldás nincs, az a hely az első chipben kihasználatlan, oda egyszerűség kedvéért ugyanaz az információ került, ami az F000:6000 és F000:C000 között van! A kompatibilis

gépek többségében nem 256, hanem 64 Kbites ROM chip-ek vannak, és akár nyolc ilyen chip is elhelyezhető. Ott a megoldás egyszerű: a kihasználatlan területre egyáltalán nem kerül információ, ott tehát a ROM terület F000:6000-en (vagy másként F600:0-on) kezdődik.

Az IBM-PC/XT interrupt vektorai

Láttuk, hogy a RAM 0:0-tól 9000:FFFF-ig terjed. Azonban ez nem áll teljes egészében a programozó rendelkezésére: az operációs rendszeren kívül vannak olyan területek, melyeken keresztül az operációs rendszer és a felhasználói programok a BIOS-szal kommunikálnak. 0:0-tól 0:3FF-ig helyezkedik el az ún. interrupt (megszakítás) táblázat. A táblázat elemei a megszakítás-vektorok, melyek mindig 4 bájtból állnak: 2 bájt szegmens és 2 bájt offset cím, melyek vagy a ROM-ba (BIOS, BASIC vagy winchester meghajtó), vagy az operációs rendszer által használt területre mutatnak.

Összesen 100(hex) ilyen interrupt vektor van, 0-tól FF-ig megszámozva. Mivel $4 \times 100 = 400$, így adódik az interrupt táblázat mérete 0:0-tól 0:3FF-ig.

Fontos megjegyezni, hogy ha egy XX interrupt ZZ:YY címre mutat, az Intel 8086 INT XX utasítása a ZZ:YY címen levő kódot hajtja végre.

A programvédelem szempontjából kulcsfontosságúak a következő megszakítások:

INT 1	(0:4 - 0:7 címen)
INT 3	(0:C - 0:F címen)
INT 8	(0:20 - 0:23 címen)
INT 9	(0:24 - 0:27 címen)
INT 13	(0:4C - 0:4F címen)
INT 1C	(0:70 - 0:73 címen)
INT 21	(0:84 - 0:87 címen)
INT 25	(0:94 - 0:97 címen)

A kulcsfontosságú megszakítások funkciója:

- Az INT 1, INT 3 megszakítást a nyomkövető (tracer, debugger) programok használják alapvetően, így ezek más célra történő felhasználása ezen programok működését zavarja.
- Az INT 9 megszakítás a billentyűzetet szolgálja ki
- Az INT 13 megszakítás a floppykezelést szolgálja ki.
- Az INT 8 (valamint az INT 1C) a belső órát (timer-t) szolgálja ki.
- Az INT 21 a DOS funkciókat kéri
- Az INT 25 a DOS közvetlen lemezkezelése (pl. Winchestereknél)

Az interrupt tábla teljes tárgyalásától itt eltekintünk, a később esetleg előforduló egyéb megszakítások működését a tárgyalási helyükön részletezzük. A teljes megszakítás tábla megtalálható az IBM XT Technical Reference-ben.

Rövid bevezetés a floppy lemezekhez

A lemez kerek plasztikdarab, rajta parányi mágneses részecskékkal. Nagyon hasonlít egy hangszalaghoz, ahol hosszú szalagcsévén helyezkednek el a részecskék. A számítógép rögzít (ír) és visszajátszik (olvas) információkat, éppen úgy, mint egy magnó, a lemez tetejére és aljára. Ebből a szempontból nagyon hasonlatos a lemezjátszóhoz, mely vagy az egyik, vagy a másik oldalát használja a lemeznek. Azonban a lemezeken van egy spirális barázda, melyet a tű követ, míg egy floppy lemezen nincs spirális barázda. Ehelyett egy számítógépes lemezen sok koncentrikus gyűrű (track) van, ahol az információ tárolódik. Egy szabványos IBM PC/XT lemezen hexadecimálisan 28, azaz decimálisan 40 ilyen track (más néven sáv) van, és mindegyik track kb. (decimálisan) 6200 bájt információt tartalmazhat. A track-ek meg vannak számozva (hexadecimálisan) 0-tól 27-ig. A lemezegységnek van egy mozgatható feje, mely egy bizonyos track-re ráállva a lemezen információt ír vagy olvas. A számítógép a lemezegységgel egy kontroller chipen keresztül kommunikál, mely kezeli a felhasználói adatok mindennemű átalakítását olyan jelekké, melyeket az olvasó/író fej használni tud.

Fontos tudnivaló, hogy bár minden lemezegység és az XT-BIOS képes kezelni a (hex.) 28-as, azaz 40. tracket (sőt, némely lemezegység és a BIOS is a 41.-42. (azaz hex. 29-2A) tracket, a PC-DOS ill. MS-DOS ezt a területet már nem kezeli, így sok védelem ezt ki is használja.

Általánosan használt szakkifejezések:

adatmező	egy szektornak az a része, mely a szektor adatait tartalmazza. Minden adatmezőnek megvan a maga CRC-je.
aldirectory	a DOS területen levő fájlok alkönyvtára
allokálás	lemezterület lefoglalás fájl számára
boot rekord	egy lemez vagy DOS partíció első szektora. Ez a szektor információt tartalmaz a lemez formátumáról, valamint tartalmazza azt a kódot is, mely az operációs rendszernek a számítógépbe való töltése közben legelőször hajtódik végre. (Ez az ún boot kód).
cylinder	ua., mint track.
cluster	A DOS által támogatott legkisebb allokálási egység. Kétoldalasan formázott floppy esetén clusterenként 2 szektor, egyoldalasra formázott floppy esetén 1 szektor.
CRC	ciklikus redundancia ellenőrzés. Olyan, a lemezen tárolt 2 bájt, melyeket arra használnak, hogy a lemez előző mezőjén levő összes bájt érvényességét ellenőrizzék velük.
CRC hiba	akkor jelentkezik, ha a kontroller, miután kiolvasta az adatmezőt, és ebből saját maga egy bizonyos algoritmus szerint CRC-t hozott létre, úgy találta, hogy az általa létrehozott CRC nem egyezik a lemezzől kiolvasott CRC-vel.
directory	a DOS területen levő fájlok könyvtára.
DOS	lemez operációs rendszer rövidítése.
DOS partíció	kemény lemezek DOS által kezelt része.

DOS terület dupla sűrűség	a DOS partíció és a meghajtóban levő teljes lemezek együtt. az a módszer, mellyel a PC kontrollere által támogatva jeleket írunk egy lemezre (néha MFM-nek is nevezik). Dupla sűrűségű módban több adat tárolódik, mint egyszeres sűrűséggel.
egyoldalás	egy lemezegység vagy lemez, mely csak egy oldalon képes információt tárolni. Amikor egy lemez leírására használják ezt a fogalmat, azt jelenti, hogy a gyártó az adathordozó közegnek csak az egyik oldalát ellenőrizte. Általában használható a nem ellenőrzött oldal is.
egyszeres sűrűség	jelírási módszer egy lemezre, néha FM-nek is nevezik. A PC controller nem támogatja információk olvasását ebben a módban.
entry FAT	bejegyzés DOS fájl allokációs táblázat. Ez a táblázat információt tartalmaz a lemez használatáról.
fej	a lemezegység által használt olvasó/író készülék, melyet a lemezeiről információ olvasására/írására használunk. Mindig egy vagy két fej van egy floppy lemezen, a sorszámuk megegyezés szerint 0 és 1, merevlemez-egységek 2-től akár 16 fejesek is lehetnek.
flag formázás	jelzőbájt adathordozón (lemezen) olyan információ elhelyezése, mely a controller számára jelzi a track-eket és szektorokat.
gap	egy lemez szektorok közti azon területe, melyet nem használnak fel a lemez írásakor és olvasásakor, hogy elkerüljék, hogy véletlenül az egyik szektorba írt adatok átlóghassanak a másik szektorba, felülírva azt.
H ID mező	a lemez feje vagy oldala, egy szektor ID mezőjében rögzített módon. egy szektornak az a része, mely a szektor címkéjét tartalmazza, beleértve a track-jét, fejét, szektor számát és szektor méretét. Minden egyes ID mezőnek megvan a saját CRC-je.
index lyuk	egy kis lyuk a lemezen, melyet a lemezegység arra használ, hogy jelezze egy track kezdetét/végét.
írásvédelem	egy kis bemetszés a lemez oldalán, melyet arra használnak, hogy a lemezegység ne írjon adatokat a lemezre, ha a bemetszés le van ragasztva.
kétoldalás	egy lemez (lemezegység), mely képes mindkét oldalán információt tárolni (írni/olvasni).
kontroller	floppy lemez kontroller vagy FDC néven is ismert. Az interface chip és a hozzá tartozó áramkörök, melyek a processzor és a lemezmeghajtó közti interface-t képezik. A kontroller feladata az, hogy adatokat írjon a lemezre, és ezeket az adatokat megbízhatóan visszavigye a számítógép memóriájába. Az IBM PC-k és vele kompatibilis számítógépek egy speciális chipet használnak a legtöbb átviteli funkció kezelésére, míg más számítógépek, mint pl. az Apple II, a számítógép processzorával végeztetik el ezt a munkát. A kontroller által véghezvitt tényleges feladat egészen komplex. Azonfelül, hogy az adatokat oda-vissza küldik, a kontrollernek meg kell találnia, hogy az adatok a lemezen hol vannak tárolva, aztán át kell alakítania a lemez fejről kapott analóg

	jeleket digitális bájtokká a számítógép számára, majd ezeket a bájtokat kell betennie a számítógép memóriájába.
logikai szektor	a DOS által alkalmazott szektor számozási rendszer. Az első logikai szektor a lemezen levő első szektor, aztán jönnek a szektorok sorban a track első oldala mentén, majd a másik oldala mentén, aztán a következő track következik.
N	egy szektor szektor méret jelzője, ahogy a szektor ID mezőjében rögzítve van. A szektorméret = $128 \times 2N$ byte. Az N szabványos értéke 2, mely decimálisan 512 bájtot jelent szektoronként.
nagy sűrűség	azon lemezek leírására használt fogalom, melyeket olyan, újabb számítógépeken használnak, mint pl. az IBM AT-n, s melyeknek több track-jük lehet, mint a normál PC lemezeknek.
oldal	a lemezegység feje(i) által használt lemezfelület (floppy lemezek esetén csak két oldal van, de winchesterek esetén 16 oldal is lehet)
pointer	mutató
rejtett szektor	ezt a fogalmat olyan szektorok leírására használják, melyek ID mezője CRC hibát tartalmaz. Ezeket a szektorokat nem találja meg a kontroller read-id parancsa.
rekord	ua., mint a szektor
root directory	egy DOS területen a fájlok főkönyvtára
S	egy szektor szektor száma, ahogy a szektor ID mezőjében rögzítve van.
szektor	egy track-en az egyik olyan terület, ahol információ van tárolva. A szektor két mezőből áll, az ID mezőből és az adat mezőből.
szektor sorszám	az a szám — a sorszámozása l-gyel kezdődik — mely egy track-en a szektorok sorrendjét jelzi, s mely általában, de nem mindig egyezik S-szel, az ID mezőben rögzített szektor számmal.
T	egy szektor track száma, ahogy ennek a szektornak az ID mezőjében rögzítve van.
track	egy lemezen egy koncentrikus gyűrű, ahol adatokat tartanak. Néha cylindernek is nevezik.
"ujjlenyomat"	olyan fogalom, mellyel előre megfontolt szándékkal létrehozott fizikai károsodást írunk le egy különben jó lemezen, s általában lézerrel való kiégetést jelent. Ezt a sérülést aztán a szoftver ellenőrzi, hogy megbizonyosodjon róla, a lemez eredeti-e.

A track-ek szerkezete

Mint láttuk, a lemez koncentrikus sávokra, ún. track-ekre oszlik. Ezek szerkezetét vizsgáljuk most. A lemezen van egy speciális lyuk, melyet index lyuknak hívnak, s melyet a kontroller arra használ, hogy ha ezt a lyukat érzékeli, tudja, hogy itt a track eleje. Mindegyik track fel van osztva szektornak nevezett részekre, melyeket tortaszeletekhez lehet hasonlítani. A számítógép által olvasott vagy írt adatok általában ezekben a szektorokban vannak tárolva.

A track-en levő információ a következőképpen van elrendezve (minden szám hexadecimális):

50*4E, 0C*00, 3*C2, FC, 32*4E, szektorok, X*4E

azaz:

50 bájtnyi 4E.

0C bájtnyi 00.

3 bájtnyi C2.

egy bájt FC.

32 bájtnyi 4E.

Következnek a szektor adatok.

Több bájtnyi 4E a track végéig.

Egy szektor formátuma:

A szektor a track egy része, ahol információ tárolódik. A szektor az a legkisebb adategység, mely lemezről beolvasható vagy arra kiíratható. Mivel a szektorokat különböző meghajtókon is tudni kell újraírni, melyek sebessége kicsit eltér egymástól, minden szektor végén szükséges egy gap, azaz rés, mely fel nem használt karakterekből áll, hogy elkerüljük, hogy az újraírás után a szektor kilógjon az eredeti helyéről. A szektorok két részből állnak: a szektor ID mezőből és az adatmezőből. A szektor ID mező akkor kerül rögzítésre, amikor a track-et először megformázzák, és sohasem kerül sor ennek újraírására. Tartalmazza az öt követő adatmező "címét", egy rövid gap után. Az adatmező tartalmazza a szektor adatait. Az adatmezőt, eltérően az ID mezőtől, mindannyiszor újra írjuk, ahányszor egy write parancsot adunk ki a lemezegység kontrollerének.

A szektor pontos szerkezete a következő:

- 0C bájtnyi 00, 3 bájtnyi A1, egy bájtnyi FE
- egy bájt track szám
- egy bájt fej (vagy oldal) szám
- egy bájt szektor ID száma
- egy bájt "N" érték, mely a szektor méretet jelzi
- az ID mező CRC két bájtja
- 16 bájtnyi 4E, 0C bájtnyi 00, 3 bájtnyi A1, egy bájtnyi FA vagy FB,
- szektor adatok
- az adatmező CRC 2 bájtja
- sok bájtnyi 4E (a gap méretétől függően)

Mindezt így lehet szemléltetni:

0C*00, 3*A1, FE, C, H, R, N, 2*CRC, 16*4E, 0C*00, 3*A1, FA, szektor, 2*CRC, X*4E

Megjegyzések:

- 1) A track szám, fej szám, szektor szám, és "N" méret értékeket közösen CHRN-nek hívják (cilinder, fej, rekord, szám angol szavak kezdőbetűi), vagy pedig THSN-nek (track, fej, szektor, "N" méret angol szavak kezdőbetűi). Az "N" értéke nem a szek-

- 2) A kétszer 3 bájtnyi A1 az az első néhány bájt, melyet a kontroller leír, amikor bekapcsolja az írófejét, hogy szektor adatokat írjon, és ezek arra a célra szolgálnak, hogy a lemez kontroller "szinkronizálja" az olvasás gyorsaságát. Ez a 3 byte ún. hiányos órabiteket tartalmaz. Ezekből a hiányzó jelekből tudja a kontroller, hogy ID vagy adatmező következik. Mivel ilyen adatmezőbe nem tudunk írni, ezért hiába írunk valamely adatszektorba a szektorleírásnak megfelelő byte-okat, a kontroller nem fogja azt hinni, hogy ez egy külön szektor!
- 3) A szektor adatok előtt közvetlenül levő FA és FB jelzi a szektor adatok kezdetét. Egy bájtnyi FA flag a szektor adatoknál azt jelenti, hogy ezek "törölt" adatok.
- 4) A gap egy olyan terület minden szektor után, melyet üresen hagynak (kezdeti formája csupa 4E) úgy, hogy a következő szektor ID-je ne hogy felülíródjon a számítógépenként kicsit eltérő meghajtó sebesség következtében. Így bár egy track több, mint decimálisan 6200 adatbájtot tartalmazhat, a számítógép csak decimálisan 4608 adatbájtot (9 darab, egyenként 512 bájtos szektort) használhat.

The diagram illustrates the layout of a disk track, showing the sequence of sectors and gaps. The track starts with a head sector (track fej) and ends with a track gap (a track maradéka). The sectors are divided into three parts: sector start (szektor kezdet), ID, and data (adat). The gaps are labeled as gap (gap) and gap length (gap hossz). The hexadecimal values for each part are as follows:

- track fej: 50*4E, 0C*00, 3*C2, FC, 32*4E
- szektor kezdet: 0C*00, 3*A1, FE, T, H, S, N, 2*CRC,
- ID: 16*4E, 0C*00, 3*A1, FB vagy FA
- adat: adatok, 2*CRC
- gap: gap hossz*4E
- a track maradéka: a track maradéka*4E

17

A DOS lemezek formátuma

A DOS lemezek jelenleg 3 féle változatban léteznek:

- DOS 1.0 verzióban 8 szektor van track-enként;
- DOS 2.0 verzióban 9 szektor van track-enként;
- DOS 3.0-ban opcionálisan a nagy sűrűségű lemezeken (decimálisan) 15 szektor van track-enként, egyébként általában 9 szektor van track-enként.

Mindhárom DOS felismeri az egy- és kétoldalas lemezeket, ill. meghajtókat. Mi a legelterjedtebb DOS 3.0 alapján fogjuk témánkat tárgyalni.

A DOS által használt mindegyik szektor 512 bájt hosszú. A cluster a DOS által használt allokáció minimális egysége, és mindig egy vagy több egymást követő szektor. A DOS terület a következő részekre osztható:

Boot rekord

fájl allokációs táblázat első clustere

fájl allokációs táblázat második clustere

root directory

adatterület a lemez végéig

Mindegyik rész mérete változó, függ a DOS verziószámától, és hogy hány oldalt használunk. A fájlok az adatterületen vannak allokálva, és csak akkor, ha szükséges (előre nincsenek allokálva). Az üres hely cluster-enként van allokálva. Egy fájl összes cluster-e a fájl allokációs táblázat szerint van egymáshoz láncolva.

A boot rekord minden, a DOS által formázott lemezen a 0-ás oldal 0-ás track-jének 1-es szektora. Avégett van minden lemezen rajta, hogy hibaüzenetet küldjön, ha a lemezen nincs rajta a rendszer is, és hogy információkat rögzítsen a lemezről. Az LS Analyse paranccsal illetve a DISK EXPLORER-rel lehet megnézni egy floppy lemez boot rekordját. (Az LS, azaz LOCKSMITH/PC és a DISK EXPLORER nevű programokról részletesebben a "Másolóprogramok és másolást segítő eszközök rövid áttekintése" c. fejezetben írunk.)

A boot rekord egymás után következő első néhány bájtjának jelentése:

00 - 02	ugrási utasítás a boot kódra
03 - 0A	a lemez vagy meghajtó gyártójának neve és verziószáma
0B - 0C	bájtok száma szektoronként
0D	szektorok száma cluster-enként
0E - 0F	fenntartott szektorok száma a lemez vagy partíció kezdetén
10	FAT-ek által elfoglalt hely. (Általában 2 clusternyi helyet foglalnak el.)
11 - 12	a fájlok száma a root directory-ban.
13 - 14	a szektorok száma összesen, egy meghajtóban
15	adathordozó közeg leíró bájt. Lehetséges értékei:
	FF kétoldalas, 8 szektoros lemez.
	FE egyoldalas, 8 szektoros lemez.
	FD kétoldalas, 9 szektoros lemez.
	FC egyoldalas, 9 szektoros lemez.

F9 kétoldalas, 15 szektoros lemez. (nagysűrűségű)

F8 merevlemez (winchester)

16 - 17 egy FAT által használt szektorok száma

18 - 19 szektorok száma track-enként

A cluster-ek a lemezen úgy vannak elrendezve, hogy a 0. lemezoldal egymás után következő szektorait használják, aztán a következő lemezoldal összes szektorát, és így tovább, amíg a track minden feje nincs allokálva. Az ezután következő használt szektor a következő track első szektora. Ez az allokációs módszer minimalizálja a fej mozgását gyorsabb lemezműveleteknél. Azonban a fájlok nem szükségszerűen vannak sorban allokálva. Az adatterület úgy allokálódik, hogy egy clustert vesz egyszerre, kezdve az első szabad cluster-nél és átugorva mindegyik allokált cluster-t. Ennek eredménye a leghatékonyabb helykihasználás a lemezen, de a fájlok nem szükségszerűen egymás utáni cluster-ek során helyezkednek el, azoknak a lyukaknak köszönhetően, melyek fájl törlések miatt keletkeztek.

A DOS root directory minden lemezen a felformázáskor létrejön (ha a DOS formázó parancsot alkalmazzuk), és csak korlátozott számú fájl bejegyzést (directory entry-t) tartalmazhat. Az alldirectory-k (alkönyvtárak) ténylegesen fájlok, és ezekben korlátlan számú fájl lehet. A directory entry-k 32 bájt hosszúak, és a következő az alakjuk (számok hexadecimálisan):

- 00-07 fájlnev: a név első bájtja jelzi az állapotszót a következőképpen:
- 00 ez után a pont után sohasem volt még fájl
 - E5 a fájl törölve van.
 - 2E ez az entry egy directory. Ha a második bájt is 2E (""), akkor ez a directory egy alldirectory-jának pointer-e. Ha a második bájt nem 2E, akkor ez az entry az aktuális alldirectory-ra mutat.
- Bármely más érték rendes fájl entry-t jelez.
- 08-0A fájlnev kiterjesztés
- 0B fájl attribútum vagy típus. A különböző, közkézen forgó segédprogramokkal meg lehet változtatni. Az egyes bitek értékei a következők:
- 01 a fájl csak olvasható
 - 02 a fájl rejtett. Nem fog megjelenni directory listánál.
 - 04 a fájl rendszer fájl.
 - 08 ez az entry a lemez címke neve. A fájlnev mező a címke, az entry többi része nem tartalmaz érvényes adatot. Csak root directory-ban engedhető meg.
 - 10 a fájl egy alldirectory
 - 20 a DOS BACKUP utasítás számára fenntartva
- 0C-15 fenntartva a későbbi fejlesztések számára
- 16-17 a fájl létrehozásának vagy legutóbbi aktualizálásának időpontja. Ebben a mezőben a bitek a következőképpen vannak felosztva:
- hhhhh mmmmmm sssss
- ahol a h jelentése óra (0-23), az m jelentése perc (0-59), és az s kétmásodperces időközt jelent (0-29).

18-19	a fájl létrehozásának vagy utolsó aktualizálásának időpontja. Ebben a mezőben a bitek a következőképpen vannak felosztva: yyyyyy mmmm dddd ahol y éveket jelent (0-119, jelentése 1980-2099), m hónapokat jelent (1-12), és d napokat jelent (1-31)
1A-1B	a fájl első cluster-e
1C-1F	fájl méret, bájtokban

A fájl allokációs táblát vagy FAT-ot a DOS arra használja, hogy a fájlok cluster-eket allokáljon. Ha kevesebb, mint (decimálisan) 4086 cluster van, minden FAT bejegyzés 12 bit hosszú, máskülönben 16 bit (csak DOS 3. és újabbak). A FAT első bájtja az adathordozó közeget leíró bájt. A következő 2 bájt (illetve 16 bites bejegyzések esetén 3 bájt) FF. Az ezután következő bájt az első FAT bejegyzés, a 002-es clusterre mutat. Az egyes bejegyzések jelentése a következő lehet, ahol a zárójelben levő szám a 16 bites FAT bejegyzésekre vonatkozik:

(0)000	a cluster nincs használva és elérhető
(F)FF7	a cluster nem használható, rossz szektort tartalmaz.
(F)FF8-(F)FFF	egy fájl utolsó cluster-e.
(X)XXX	ez a cluster el van foglalva, és arra a következő cluster-re mutat, melyet ez a fájl használ.

Egy cluster-számból a logikai szektor számot a következő képlettel számíthatjuk:

logikai szektorszám = (cluster-2)*szektorok clusterenként - első log.szektor

.EXE és .COM fájlok a DOS alatt

A DOS operációs rendszerben alapvetően három típusú fájl hajtható végre úgy, hogy a nevét egyszerűen begépeljük:

- .BAT kiterjesztésű fájlok. Ezek közönséges szövegfájlok, minden soruk egy DOS parancsból vagy más végrehajtható fájlból áll.
- .COM kiterjesztésű fájlok. Ezek már bináris fájlok, tehát a processzor kódot tartalmaz. Méretük nem lehet 64 K-nál nagyobb és történelmileg a régebbi CP/M operációs rendszer .COM fájljaira vezethetők vissza, a 8086 (illetve 80186, 80286, 80386)-os processzor által biztosított lehetőségeket nem használják ki. Fejlécük ("Header") nincs, a processzor rögtön az első bájtól kezdi őket végrehajtani, miután mindig a felhasználói programok számára első szabad szegmens hex. 100-as címére töltődnek.
- .EXE kiterjesztésű fájlok esetén nincs 64 K-s korlát, ezek a tipikus fájlok a PC/MS-DOS-ban. A .COM fájlaktól eltérően az első két bájtjuk mindig "MZ", ezután egy hosszabb header következik, mely egyéb információt is közöl (tényleges hossz, start cím, start szegmensregiszter, ún. relokációs faktorok táblázata, mely táblázat értékei – mivel a felhasználói programok számára első szabad szegmens sokszor más és más – az ennek függvényében változó, .EXE fájlban levő bájtokra mutatnak).

II. A másolóprogramok és másolást segítő eszközök rövid áttekintése

Az IBM operációs rendszeréhez (DOS) néhány segédprogram is tartozik, egyik közülük az ún. DISKCOPY. Minden olyan programot (pontosabban olyan lemezt, melyen az illető program bizonyos része található) másolásvédeletnek kell tekinteni, melyet a DISKCOPY vagy a DOS COPY parancs nem tud úgy lemásolni, hogy a másolat funkcionálisan megfeleljen az eredetinek.

Alapvető tendencia, hogy a különböző, DISKCOPY-t helyettesítő ún. másolóprogramok és az egyre újabb másolásvédelmek egymással versengve fejlődnek.

Míg azonban védelmeket nagyon sok cég fejleszt, addig a PC számára készült másolóprogramokkal alig féltucat gyártó rukkolt ki, így a helyzet merőben más, mint a Commodore 64 "világában". A gyártók becsületére legyen mondva, hogy a másolóprogramok az IBM PC-re többnyire jobban sikerültek, mint a védelmek. Magyarországon a legelterjedtebb két másolóprogram a COPYWRIT és a COPY II PC. Sokan használják másolásra a LOCKSMITH/PC-t (a LOCKSMITH az APPLE II számítógép kedvelőit hódította meg először) és segít a másolásban a DISK EXPLORER is, mely általános floppy lemezvizsgáló program (a LOCKSMITH-nek is van lemezvizsgáló opciója). Mivel ezeket a programokat a gyártók állandóan továbbfejlesztik, s a kibocsátott új meg új változatok viharesebesen terjednek el a magyarországi felhasználók körében, a másolásvédelem magyar fejlesztőinek ugyanúgy számolni kell velük, mint a külföldi fejlesztőknek.

Alapvetően hibás szemléletet tükröz, hogy a laikusok a másolóprogramokat "bitenkénti copy"-nak vagy "bit-copy"-nak nevezik el. "Bitenkénti" másolóprogram, mely a lemezről "tükörmásolatot" csinál, nincs, sőt elméletileg sem létezhet. Még az újabban terjedő COPY II OPTION BOARD bővítőkártya, mely kiváltja a PC eredeti floppy controllerét, sem képes a lemezek abszolút biztos másolására. Egyszerű ellenpélda: már az, ha szándékosan fizikailag felsértik a lemez mágneses rétegét, lehetetlenné teszi a tükörmásolást, de tulajdonképpen erre sincs szükség. Ugyanis nincs két egyforma floppy meghajtó, még ha azonos típusúakat hasonlítunk is össze. Elméletileg sem lehet két, teljesen azonos meghajtót legyártani, így a track-ek teljes hossza meghajtóspecifikus. A track-ek hosszának a betartását a COPY II board meghajtóprogramjától lehet kérni, de az nem hajtja (valószínűleg szándékosan) azt korrekt módon végre. Ezt néhány új védelem (SUPERLOK 3.0, EVERLOK) ki is használja.

A másolás egy másik módja az, hogy — mivel esetleg úgysem lehet a másolatot tökéletesen elkészíteni — olyan kicsi, memóriarezidens (a gép tárában "ülő") programokat készítenek, melyek azáltal, hogy különböző interruptok működését futás közben szabályozzák, ha felismerik az éppen futó védelmet, nem engedik azt érvényesülni. Ezeket a programokat általában a másolóprogramokhoz szokták mellékelni, így ezekből a folyamatos "ellátás" biztosítva van. Csak az ismertebbeket említem: a ZERODISK, RAMKEY, UNGUARD — melyek a COPYWRIT tartozékai, s a NOKEY és a NOGUARD — a COPY II PC szerves részei. Egészen szélsőséges tulajdonságai vannak ezek közül a NOGUARD-nak és az UNGUARD-nak: bizonyos védelmekről az illető programokat örökre megszabadítják: egyszer lefuttatva velük a védett programot, a másolásvédelem volt — nincs.

A védelem feltörésére szolgáló harmadik módszer a legbiztosabb és gyakorlatilag kivédhetetlen: kellő tapasztalattal, ismeretekkel (és tegyük hozzá: programeszközökkel, segédeszközökkel) rendelkező szakember bármely védelmet képes hosszabb – rövidebb idő alatt a programról eltávolítani, s azt egyszer és mindenkorra másolhatóvá tenni. A védelem ilyen feltörését a védelmet gyártó cégek megpróbálják megnehezíteni, ismert módszer például a nyomkövető programok (debuggerek) akadályozása az INT 1 és INT 3 interrupt vektorok átírása révén (azaz a 0:4 – 0:7-ig és a 0:C – 0:F területek felülírása), sőt néha odáig elmennek, hogy az egész interrupt skálát felülírják saját kóddal, majd az ellenőrzés után a védelem visszállítja az eredeti interrupt táblát, miközben a program természetesen állandóan változtatja önmagát. A védők és másolók közti erőviszonyok azonban nem egyenlők:

- a megfejtők számára korlátlan idő áll rendelkezésre, a védelmet készítő szakember számára nem;
- a védelmet készítő szakemberekkel szemben a felhasználók lényegesen nagyobb tábora áll szemben.

Így tulajdonképpen a legbiztosabb másolásvédelem a jól eltalált fogyasztói ár: sokszor drágább a szoftver felhasználásához nélkülözhetetlen dokumentációt fénymásolni, mint az eredeti terméket megvenni. (Ravasz módszer az is, hogy a dokumentációt olyan nyomda-technikai bevonattal látják el, hogy a közönséges fénymásoló gépekkel a jámbor másoló kudarcot vall; azonban ez a nehézség is áthidalható.)

Meg kell említeni még a másolásvédelem speciális eseteit, a hardveres védelmet, amikor a különleges információt nem a floppy lemezekre viszik rá, hanem speciális bővítőkártyák – ún. "kulcskártyák" – vagy egyéb, a gépben szabványos módon bentlevő interface-ekre rácsatlakoztatható "fekete dobozok" formájában adják a programcsomagokhoz (pl. a NOVELL NetWare vagy az AUTOCAD 2.50 programcsomag). Ezekben az esetekben a másolás csak az előbbi harmadik módszerrel lehetséges, vagyis egy tapasztalt szakembernek kell megszabadítania a programot a védelemtől.

Hogyan működik a másolás elleni védelem?

Egy szabványos, DOS 2.0-ás vagy újabb verziószámú lemezen 9 szektor van track-enként, 1-től 9-ig. Mindegyik szektorra "T" az az aktuális fizikai track, melyen a szektor ténylegesen rajta van, a "H" a fej, lehet 0 vagy 1, és az "N" értéke 2, azaz (decimálisan) 512 bájtnyi adat. Mindaz, amit egy védelmi séma tesz, csak annyi, hogy keres valamit, ami különleges, egy bizonyos track-en vagy track sorozaton.

A másolásvédelem hosszú utat járt be a kezdetek óta, és most számos eljárást használ. Ezek közül bemutatunk néhány elterjedtebb típust.

Egyszerű módszer az, hogyha egyszerűen megváltoztatják az 1-től 9-ig terjedő szektor sorszámozást valami másra, például 1,9,2,8,3,7,4,6,5. A számítógépnek csak be kell olvasnia az adatokat a kontroller több szektoros opciója segítségével, és meg kell néznie, hogy az adatok elcsúsztak-e. Ha a lemezt a szokásos DISKCOPY-val másolták le, az adatok nem lesznek ugyanabban a sorrendben a másolaton, mint az eredetin. A program

ekkor észrevenné, hogy ez egy "hamisítvány". Ennek a módszernek egy másik változata az, hogy egy szektornak nem szokásos sorszámot adunk, pl. a szektorok így következnenek a track-ben:

1,18,3,4,5,6,7,8,9.

A DISKCOPY ezt nem tudja lemásolni, a prógramból pedig nagyon egyszerűen rá lehet kérdezni a 18-as számú szektor meglétére.

A védelem másik módszere azt alkalmazza, hogy a lemezt úgy formázza, hogy pontosan meghatározott számú bájt legyen a gap-ben. Szabályos DOS formátumban (decimálisan) 80 bájt van a szektorok közti gap-ben.

Ha egy track-et 9 darab szektorra formázunk ahol $N=2$, de az egyik szektort $N=3$ -ként címkézzük meg, lehetséges információkat beolvasni az adatbájtok utáni részről is, vagyis ténylegesen beolvasni a CRC bájtokat, a gap bájtokat, és a következő szektor elejét, beleértve annak ID-jét. Ez azért működik így, mert a track formázásakor a format parancs (decimálisan) 512 adat bájtot allokal, mindegyik szektor adat részére, de amikor az adatokat visszaolvassa, a parancs (decimálisan) 1024 bájtot allokal, ahogy a szektor ID-jében specifikálva van. Ez a védelmi módszer egy kezdő számára nagyon rejtélyes, de sok jel van, mely megmutatja, hogyha ezt a fajta módszert használják. Az egyik jel az, hogy egy olyan szektor, melynek N -je 1024 bájtot jelez, a lemezen levő hely ugyanakkora százalékat foglalja el, mint egy csak 512 bájtos méretű szektor. Ennek egy mellékterméke lesz egy adat CRC hiba a szektorban. Ennek magyarázata az, hogy a szektort úgy formázták, hogy CRC bájtjai a rendes $N=2$ -s szektornak megfelelő helyen voltak, de amikor a read-et hajtja végre a kontroller, ezeket csak rendes adatokként olvassa be.

Gyakran a védelmi sémák szándékos CRC hibákkal írnak szektorokat. Ha az adatrészben van CRC hiba, a legtöbb másolóprogram könnyen megbirkózik az ilyen védelemmel.

Ha az ID-ben van egy CRC hiba, a read-id parancs nem fogja jelezni a szektor létét. Ez az, amit "rejtett" szektornak hívnak, mivel láthatatlan a rendes szektor jelzési eljárások számára. De a szektor információ más módszerekkel is megtalálható. A track read paranccsal a track egész tartalma vizsgálható. Ha a track read parancsot $N=6$ értékkel adjuk ki, (mely decimálisan 8192 adat bájtot jelent), az adatokat az első szektornál kezd el beolvasni, és az egész track-et beolvassa, mintha egy hatalmas, 8K-s szektort olvasna. Mivel csak kb. 6200 bájt van egy track-en, a track első részét kétszer fogja beolvasni. Aztán meg lehet nézni az adat puffert, és megkeresni a szektor ID-jét. Sajnos ezzel kapcsolatban problémák merülnek fel: Amikor a track-et formázták, az összes írást a lemezre egy műveletben hajtották végre. Ennek eredménye olyan track lett, melyet könnyű vizsgálni. Amikor egy szektort írunk, annak felül kell írnia a régi adatokat. Az adatok néhány bittel eltérhetnek, például azt okozva, hogy egy A4 (mely binárisan 10100100) úgy fog kinézni, mint egy 29 (binárisan 00101001). Azonban a puffert félbitenként is lehet vizsgálni, s így a lemez időzítési információ fog megjelenni a tényleges adatok helyett. Ezt egyetlen módon lehet helyrehozni: addig olvastatni újra a track-et, míg a kívánt adatok nem "szinkronban" jelennek meg. Egy track elején levő adatok kiolvasása is nehézséget okozhat, mivel a format parancs leáll, amikor eléri az index lyukat, s még egy visszacsatolást sem hoz létre a track elejére. Emiatt a probléma miatt sok rejtett vizsgálatba kerülhet, amíg megtaláljuk egy track-re a helyes szektor ID-ket.

Másolásvédelmi rendszerek

PROLOK

A továbbiakban két ismertebb másolásvédelmet tekintünk át, melyeknek közös jellemzője, hogy egyszerű eszközökkel nem másolhatóak le és viszonylag jól megoldották fejlesztőik a debug (nyomkövető) programok hatástalanítását is. Korábbi és talán ismertebb a Vault Corporation által készített Program Lock ("programzár"), rövid nevén a PROLOK (és nem PROLOCK!). Magyarországra többnyire a népszerű dBASE III szoftver által került be, ugyanis a dBASE szerzői ezzel a védelemmel látták el terméküket. Ez a védelem először mindenki számára szokatlan volt: a programlemezen volt ugyanis a (decimálisan) 39. track-en, a 0. lemezoldalon egy szemmel is jól látható, lézerrel kiégetett lyuk. A programot a DOS "COPY" parancsával akárhova lehetett másolni, indításkor azonban a gép lemezneghajtójába be kellett helyezni a "lyukas" lemezt, írásvédelem nélkül. A PROLOK ugyanis felülírta a 39. tracknek a lyukat tartalmazó szektorát, majd elolvasta. Ha az olvasás után nem jelentkezett adat CRC hiba, felismerte a hamisítványt, mivel a másolóprogramok a lézer lyuk okozta CRC hibát átmásolták ugyan, de lyukat égetni nem tudtak. Az ép lemezfelületen levő CRC hiba adatfelíráskor megsemmisült. Több hiábavaló próbálgatás után a bátrabb felhasználók megpróbálták DOS debug vagy más nyomkövetők segítségével kideríteni, hogy hol van a védelem elrejtve. Ekkor jött a másik meglepetés: a PROLOK futását nem lehetett a szokásos módon nyomkövetni. Egy-két szakember ennek ellenére, több heti vagy hónapi munkával sikerrel járt, de mint ilyenkor szokás, az általuk megszerzett információt gondosan eltitkolták.

A PROLOK előnyei szembetűnőek:

- A védelem gyártása és a program gyártása jól elkülönül, a védelem bármelyik programra ráinstallálható néhány másodperc alatt, s az installálást még laikus is elvégezheti.
- A nagy tömegben gyártott, kiégetett lemezek viszonylag olcsóak.
- A lézer égette lyuk a 39. track-en véletlenszerűen helyezkedik el, így minden levédett program egy kicsit más, lehetőség van a kalózpéldányok követésére ("ujjlenyomat"-módszer).
- Jól elkészített, kellően átláthatatlan, nehezen nyomkövethető kód.

A PROLOK hátrányai csak valamivel később derültek ki:

- A különböző memóriarezidens programok számára meghagyta a lehetőséget az interruptok, mindenekelőtt az INT 13 átirányítására. A RAMKEY, NOKEY, ZERODISK ezt használja ki, és így bár a COPYWRIT és a COPY II PC nem képes a lézerlyukas lemezt lemásolni, a velük egy csomagban szállított utility-k (segédprogramok) egyszerűen megkerülik a védelmet.
- A felhasználók félnek a lézer által égetett lyuktól, hogy annak szélei esetleg túlzottan elkoptatják a floppy meghajtó író/olvasó fejét.
- Az ügyes kezű felhasználó maga is imitálhatja a lézer által égetett lyukat úgy, hogy a megfelelő helyen megkarcolja a lemezt. Ez a módszer még eléggé biztosnak is mondha-

tó, mivel a PROLOK csak a lyuk kezdetét ellenőrzi, s így a lemez forgásával szemben haladva, milliméterenként meghosszabbítva a karcolást és állandóan ellenőrizve a PROLOK futását, néhány percen belül a "lézeres" lemez előállítható.

A legnagyobb probléma az, hogy egy kis ügyességgel (vagy a megfelelő segédprogram birtokában) a PROLOK nyomkövetése debuggerrel meglehetősen egyszerű: nem csak az INT 13 megszakításról "feledkeztek" meg a gyártók, hanem az INT 9-ről, a billentyűzet megszakításáról is. Néhány ismert debugger (SST +, AFD) az INT 9-et magára átirányítja, és ezek a debuggerek még memóriarezidenssé is tehetők. Ezután csak a felhasználó reflexeinek gyorsaságán múlik az egész. Ha ugyanis kellő időben lenyom egy billentyűt, a PROLOK a debuggerhez tér vissza, s azután rövid és egyszerű nyomkövetés végeztével a memóriában teljes egészében látható lesz a PROLOK által levédett program (pl. dBASE-III) a védelem gyári installálása előtti állapotában. Ha ezt a memóriarészt kimentjük egy lemezre, a PROLOK lemezre soha nem lesz szükség.

Így ez az INT 9 megszakítás rengeteg kellemetlenséget okozhat egy védelem eladóinak. Furcsa módon mintha megfeledkeztek volna róla a nyugati gyártók: Magyarországra eddig még egyetlen egy olyan "védett" nyugati termék (beleértve magukat a védelmeket) nem került be, mely az INT 9-et hatástalanította volna.

A felhasználók panaszai miatt – hogy a lyuk tönkreteszi a meghajtó fejét, s a PROLOK a winchesterre nem installálható (ha PROLOK-kal védett lemezünk van, nem tudjuk a programot csak winchesterről indítani, s a felhasználók és a gyártó szerint ha már valaki "védett" terméket használ, ne kelljen állandóan ki-be dugdosnia a PROLOK-lemezt, inkább a védelem valahogy "mentődjön át" a winchesterre) – a dBASE gyártója, az ASHTON-TATE cég később lemondott a PROLOK használatáról, s nyomában sok más gyártó is (a LOTUS, BMDP, stb programcsomagok gyártói) áttért a SOFTGUARD cég SUPERLOCK védelmére, mely kb. másfél éven keresztül a legelterjedtebb védelem volt.

A PROLOK-nak és a VAULT CORPORATION-nak azonban még nem volt vége. A PROLOK különböző verziói még mindig léteznek a piacon, s folyamatos javítások, toldozások eredményeként már nem lehet egyszerűen tűvel vagy szikével felsérteni a lemezt, a lézerlyukat utánozandó. (Az új PROLOK, a PROLOK-2 egy helyett ugyanis két, egymás melletti lézerlyukat használ, és jóval pontosabban ellenőrzi a védelem ezek helyét), de a PROLOK-2 változatlanul megfeledkezik az INT 9 megszakításról, s így kb. 5-10 perc alatt feltörhető.

A PROLOK-nak számos érdekes hazai vonatkozása is volt:

- A lézerlyukakat kísérő misztika, titkolózás sok embert izgatott, s izgat talán most is. Kezdetben az a tény, hogy valaki tudta másolni (esetleg visszainstallálni) a PROLOK védelmet, piaci előnyöket és tekintélyt adott.
- Tanui lehettünk a magyar PROLOK keletkezésének. Több különböző, CRYPTO-nak nevezett védelem verzió jött létre. Aki tüzetesebben megvizsgálja a CRYPTO típusú védelmeket, rájön, hogy azokban többé-kevésbé módosított PROLOK kód van. Bizonyágképpen íme a CRYPTO, a PROLOK 1.07 és a PROLOK 2.0 főciklusa, amelyek lényegében nem térnek el egymástól.

1D8D:1B60	2E891EAB16	mov	cs:[16AB],bx
1D8D:1B65	2E8B1EA516	mov	bx,cs:[16A5]
1D8D:1B6A	2E891EA716	mov	cs:[16A7],bx
1D8D:1B6F	D1E3	shl	bx,1
1D8D:1B71	D1E3	shl	bx,1
1D8D:1B73	D1E3	shl	bx,1
1D8D:1B75	D1E3	shl	bx,1
1D8D:1B77	2E89872E16	mov	cs:[bx+162E],ax
1D8D:1B7C	2EA1AB16	mov	ax,cs:[16AB]
1D8D:1B80	2E89873016	mov	cs:[bx+1630],ax
1D8D:1B85	2E898F3216	mov	cs:[bx+1632],cx
1D8D:1B8A	2E89973416	mov	cs:[bx+1634],dx
1D8D:1B8F	2E89B73616	mov	cs:[bx+1636],si
1D8D:1B94	2E89BF3816	mov	cs:[bx+1638],di
1D8D:1B99	2E8C9F3A16	mov	cs:[bx+163A],ds
1D8D:1B9E	2E8C873C16	mov	cs:[bx+163C],es
1D8D:1BA3	D1EB	shr	bx,1
1D8D:1BA5	D1EB	shr	bx,1
1D8D:1BA7	D1EB	shr	bx,1
1D8D:1BA9	2E8F876E16	pop	cs:[bx+166E]
1D8D:1BAE	2E8F06AB16	pop	cs:[16AB]
1D8D:1BB3	2E8F877616	pop	cs:[bx+1676]
1D8D:1BB8	83C302	add	bx,+02
1D8D:1BBB	83FB08	cmp	bx,+08
1D8D:1BBE	7202	jc	1BC2
1D8D:1BC0	33DB	xor	bx,bx
1D8D:1BC2	D1EB	shr	bx,1
1D8D:1BC4	2E891EA516	mov	cs:[16A5],bx
1D8D:1BC9	0E	push	cs
1D8D:1BCA	1F	pop	ds
1D8D:1BCB	FC	cld	
1D8D:1BCC	B9B41C	mov	cx,1CB4
1D8D:1BCF	BE601B	mov	si,1B60
1D8D:1BD2	2BCE	sub	cx,si
1D8D:1BD4	33D2	xor	dx,dx
1D8D:1BD6	D1E9	shr	cx,1
1D8D:1BD8	AD	lodsw	
1D8D:1BD9	03D0	add	dx,ax
1D8D:1BDB	E2FB	loop	1BD8
1D8D:1BDD	2EA1E81B	mov	ax,cs:[1BE8]
1D8D:1BE1	03C0	add	ax,ax
1D8D:1BE3	3BD0	cmp	dx,ax
1D8D:1BE5	7403	jz	1BEA
1D8D:1BE7	F4	hlt	
1D8D:1BE8	FD	std	
1D8D:1BE9	E0FB	loopnz	1BE6
1D8D:1BEA	FB	sti	
1D8D:1BEB	2EFF06A916	inc	w,cs:[16A9]
1D8D:1BF0	E802F8	call	13F5
1D8D:1BF3	817F04601B	cmp	w,[bx+04],1B60
1D8D:1BF8	75071EAB16	jnz	1C01
1D8D:1BFA	8CC8	mov	ax,cs
1D8D:1BFC	394706	cmp	[bx+06],ax
1D8D:1BFF	7402	jz	1C03
1D8D:1C01	FA	cli	
1D8D:1C02	F4	hlt	
1D8D:1C03	0E	push	cs

1D8D:1C04	1F	pop	ds
1D8D:1C05	89269703	mov	[397],sp
1D8D:1C09	B90900	mov	cx,9
1D8D:1C0C	A1A916	mov	ax,[16A9]
1D8D:1C0F	BBFFFF	mov	bx,0FFFF
1D8D:1C12	43	inc	bx
1D8D:1C13	3A877E16	cmp	al,[bx+167E]
1D8D:1C17	EOF9	loopnz	1C12
1D8D:1C19	754A	jnz	1C65
1D8D:1C1B	50	push	ax
1D8D:1C1C	53	push	bx
1D8D:1C1D	51	push	cx
1D8D:1C1E	D1E3	shl	bx,1
1D8D:1C20	8B878716	mov	ax,[bx+1687]
1D8D:1C24	3D0000	cmp	ax,0
1D8D:1C27	7432	jz	1C5B
1D8D:1C29	FC	cld	
1D8D:1C2A	0E	push	cs
1D8D:1C2B	1F	pop	ds
1D8D:1C2C	1E	push	ds
1D8D:1C2D	07	pop	es
1D8D:1C2E	8BF0	mov	si,ax
1D8D:1C30	AD	lodsw	
1D8D:1C31	8936AD16	mov	[16AD],si
1D8D:1C35	8BC8	mov	cx,ax
1D8D:1C37	49	dec	cx
1D8D:1C38	AC	lodsb	
1D8D:1C39	8BFE	mov	di,si
1D8D:1C3B	8AD0	mov	dl,al
1D8D:1C3D	AC	lodsb	
1D8D:1C3E	86D0	xchg	dl,al
1D8D:1C40	02C2	add	al,dl
1D8D:1C42	AA	stosb	
1D8D:1C43	E2F8	loop	1C3D
1D8D:1C45	A1AD16	mov	ax,[16AD]
1D8D:1C48	FFD0	call	ax
1D8D:1C4A	8B3EAD16	mov	di,[16AD]
1D8D:1C4E	83EF02	sub	di,+02
1D8D:1C51	0E	push	cs
1D8D:1C52	07	pop	es
1D8D:1C53	FC	cld	
1D8D:1C54	8B0D	mov	cx,[di]
1D8D:1C56	49	dec	cx
1D8D:1C57	F3AA	repz	stosb
1D8D:1C59	EBC9	jmps	1C24
1D8D:1C5B	59	pop	cx
1D8D:1C5C	5B	pop	bx
1D8D:1C5D	58	pop	ax
1D8D:1C5E	0E	push	cs
1D8D:1C5F	1F	pop	ds
1D8D:1C60	C6877E1600	mov	b,[bx+167E],0;" "
1D8D:1C65	E302	jcxz	1C69
1D8D:1C67	EBA9	jmps	1C12
1D8D:1C69	FA	cli	
1D8D:1C6A	33DB	xor	bx,bx
1D8D:1C6C	8EC3	mov	es,bx
1D8D:1C6E	26C74704601B	mov	w,es:[bx+04],1B60

1D8D:1C74	8B1EA516	mov	bx,[16A5]
1D8D:1C78	D1E3	shl	bx,1
1D8D:1C7A	FFB77616	push	[bx+1676]
1D8D:1C7E	0E	push	cs
1D8D:1C7F	FFB76E16	push	[bx+166E]
1D8D:1C83	FB	sti	
1D8D:1C84	B103	mov	cl,3
1D8D:1C86	D3E3	shl	bx,cl
1D8D:1C88	8E873C16	mov	es,[bx+163C]
1D8D:1C8C	8BBF3816	mov	di,[bx+1638]
1D8D:1C90	8BB73616	mov	si,[bx+1636]
1D8D:1C94	8B973416	mov	dx,[bx+1634]
1D8D:1C98	8B8F3216	mov	cx,[bx+1632]
1D8D:1C9C	8B873016	mov	ax,[bx+1630]
1D8D:1CA0	A3AB16	mov	[16AB],ax
1D8D:1CA3	8B872E16	mov	ax,[bx+162E]
1D8D:1CA7	8E9F3A16	mov	ds,[bx+163A]
1D8D:1CAB	2E8B1EAB16	mov	bx,cs:[16AB]
1D8D:1CB0	CF	iret	

Figyelemre méltóak itt a következő trükkök:

- 1BD8 és 1BDB között az egész rutinról "kontrollösszeg" készül. A debug programok viszont egységesen úgy működnek, hogy a töréspontokra hex. CC-t helyeznek, így ha a töréspontot a kontrollösszeg képzése után helyezzük el, ez hibás kontrollösszeget fog eredményezni.
- 1BF3 és 1C02 között a 0:4-en levő információ ellenőrzése történik. Ha az nem CS:1B60, akkor CLI, HLT utasítás-sorozat (megszakítások letiltása, processzor állj) következik.
- A leginkább figyelemre méltó az, hogy a rutin legvégén a 0:4-be mindig CS:1B60-t tölt, kilépés IRET-tel történik, ami a régi flag-értékeket betölti a STACK tetejéről, ott a trap flag értéke 1, és így az INT1 hajtódik végre, azaz egy indirekt ugrás 1B60-ra.

Megjegyzés: az 1BF0-án levő CALL a következő rutint hívja:

1D8D:13F5	push	cs
1D8D:13F6	pop	ds
1D8D:13F7	push	ds
1D8D:13F8	pop	es
1D8D:13F9	cld	
1D8D:13FA	mov	bx,[16A7]
1D8D:13FE	shl	bx,1
1D8D:1400	mov	di,[bx+166E]
1D8D:1404	sub	di,+07
1D8D:1407	mov	cx,6
1D8D:140A	mov	al,41 ; "A"
1D8D:140C	inc	al
1D8D:140E	stosb	
1D8D:140F	loop	140C
1D8D:1411	mov	bx,[16A5]
1D8D:1415	mov	dl,[bx+1699]
1D8D:1419	shl	bx,1

```

1D8D:141B    mov     di,[bx+169D]
1D8D:141F    mov     si,di
1D8D:1421    mov     cx,si
1D8D:1423    sub     cx,[bx+166E]
1D8D:1427    xchg    ch,cl
1D8D:1429    mov     cl,7
1D8D:142B    sub     cl,ch
1D8D:142D    and     cx,7
1D8D:1431    lodsb
1D8D:1432    xor     al,dl
1D8D:1434    stosb
1D8D:1435    loop    1431
1D8D:1437    mov     [bx+169D],di
1D8D:143B    xor     bx,bx
1D8D:143D    mov     ds,bx
1D8D:143F    ret

```

Továbbá lássuk a PROLOK 1.07 kódját:

```

1D8D:1E50    mov     cs:[170E],bx
1D8D:1E55    mov     bx,cs:[16BA]
1D8D:1E5A    mov     cs:[16BC],bx
1D8D:1E5F    shl     bx,1
1D8D:1E61    shl     bx,1
1D8D:1E63    shl     bx,1
1D8D:1E65    shl     bx,1
1D8D:1E67    mov     cs:[bx+1D3A],ds
1D8D:1E6C    mov     cs:[bx+1D3C],es
1D8D:1E71    mov     cs:[bx+1D3E],ax
1D8D:1E76    mov     ax,cs:[170E]
1D8D:1E7A    mov     cs:[bx+1D40],ax
1D8D:1E7F    mov     cs:[bx+1D42],cx
1D8D:1E84    mov     cs:[bx+1D44],dx
1D8D:1E89    mov     cs:[bx+1D46],si
1D8D:1E8E    mov     cs:[bx+1D48],di
1D8D:1E93    shr     bx,1
1D8D:1E95    shr     bx,1
1D8D:1E97    shr     bx,1
1D8D:1E99    pop     cs:[bx+16AA]
1D8D:1E9E    pop     cs:[170E]
1D8D:1EA3    pop     cs:[bx+16B2]
1D8D:1EA8    add     bx,+02
1D8D:1EAB    cmp     bx,+08
1D8D:1EAE    jc      1EB2
1D8D:1EB0    xor     bx,bx
1D8D:1EB2    shr     bx,1
1D8D:1EB4    mov     cs:[16BA],bx
1D8D:1EB9    sti
1D8D:1EBA    push    cs
1D8D:1EBB    pop     ds
1D8D:1EBC    inc     w,[16CA]
1D8D:1EC0    pop     ds
1D8D:1EC1    push    ds
1D8D:1EC2    xor     ax,ax
1D8D:1EC4    mov     es,ax
1D8D:1EC6    mov     di,0

```

```

1D8D:1EC9      mov          si, bp
1D8D:1ECB      lodsw
1D8D:1ECC      stosw
1D8D:1ECD      mov          bx, cs
1D8D:1ECF      mov          ax, bx
1D8D:1ED1      stosw
1D8D:1ED2      add          di, +04
1D8D:1ED5      mov          cx, 3
1D8D:1ED8      test         b, cs:[61C], 18
1D8D:1EDE      jz           1EE4
1D8D:1EE0      add          di, +04
1D8D:1EE3      dec          cx
1D8D:1EE4      lodsb
1D8D:1EE5      stosb
1D8D:1EE6      lodsb
1D8D:1EE7      stosb
1D8D:1EE8      mov          ax, cs:[16CA]
1D8D:1EEC      and          ax, 7
1D8D:1EEF      add          bx, ax
1D8D:1EF1      mov          es:[di], bx
1D8D:1EF4      add          bx, cx
1D8D:1EF6      add          di, +02
1D8D:1EF9      loop         1EE4
1D8D:1EFB      push         cs
1D8D:1EFC      pop          ds
1D8D:1EFD      call         1F8B
1D8D:1F00      call         1FB1
1D8D:1F03      cmp          w, [4], 1E50
1D8D:1F09      jz           1F0D
1D8D:1F0B      cli
1D8D:1F0C      hlt
1D8D:1F0D      push         cs
1D8D:1F0E      pop          ds
1D8D:1F0F      mov          ax, [16CA]
1D8D:1F12      mov          cx, 16
1D8D:1F15      mov          bx, 0FFFF
1D8D:1F18      inc          bx
1D8D:1F19      cmp          al, [bx+16CC]
1D8D:1F1D      loopnz      1F18
1D8D:1F1F      jnz         1F3D
1D8D:1F21      push         ax
1D8D:1F22      push         bx
1D8D:1F23      push         cx
1D8D:1F24      shl          bx, 1
1D8D:1F26      call        [bx+16E2]
1D8D:1F2A      cmp          ax, 0
1D8D:1F2D      jz           1F33
1D8D:1F2F      call        ax
1D8D:1F31      jmps        1F2A
1D8D:1F33      pop          cx
1D8D:1F34      pop          bx
1D8D:1F35      pop          ax
1D8D:1F36      push        cs
1D8D:1F37      pop          ds
1D8D:1F38      mov          b, [bx+16CC], 0
1D8D:1F3D      jcxz        1F41
1D8D:1F3F      jmps        1F18

```



```

1D8D:1F41    xor      ax,ax
1D8D:1F43    mov      es,ax
1D8D:1F45    mov      w,es:[4],1E50
1D8D:1F4C    mov      bx,[16BA]
1D8D:1F50    shl      bx,1
1D8D:1F52    push     cs:[bx+16B2]
1D8D:1F57    push     cs
1D8D:1F58    push     cs:[bx+16AA]
1D8D:1F5D    mov      cl,3
1D8D:1F5F    shl      bx,cl
1D8D:1F61    mov      es,[bx+1D3C]
1D8D:1F65    mov      ax,[bx+1D3E]
1D8D:1F69    mov      cx,[bx+1D40]
1D8D:1F6D    mov      [170E],cx
1D8D:1F71    mov      cx,[bx+1D42]
1D8D:1F75    mov      dx,[bx+1D44]
1D8D:1F79    mov      si,[bx+1D46]
1D8D:1F7D    mov      di,[bx+1D48]
1D8D:1F81    mov      ds,[bx+1D3A]
1D8D:1F85    mov      bx,cs:[170E]
1D8D:1F8A    ired

```

Az 1EFD-n és az 1F00-án levő CALL a következő rutinokat hívja meg:

```

1D8D:1F8B    mov      bx,cs:[16BC]
1D8D:1F8B    mov      bx,cs:[16BC]
1D8D:1F90    shl      bx,1
1D8D:1F92    mov      di,[bx+16AA]
1D8D:1F96    sub      di,+07
1D8D:1F99    mov      si,di
1D8D:1F9B    mov      ax,cs
1D8D:1F9D    mov      es,ax
1D8D:1F9F    mov      ds,ax
1D8D:1FA1    mov      cx,5
1D8D:1FA4    lodsb
1D8D:1FA5    xor      al,bl
1D8D:1FA7    mov      bl,al
1D8D:1FA9    add      bx,[16CA]
1D8D:1FAD    stosb
1D8D:1FAE    loop     1FA4
1D8D:1FB0    ret

```

Illetve:

```

1D8D:1FB1    push     cs
1D8D:1FB2    pop      ds
1D8D:1FB3    mov      bx,[16BA]
1D8D:1FB7    mov      dl,[bx+16C6]
1D8D:1FBB    shl      bx,1
1D8D:1FBD    mov      di,[bx+16BE]
1D8D:1FC1    mov      si,di
1D8D:1FC3    mov      ax,di
1D8D:1FC5    push     cs
1D8D:1FC6    pop      es
1D8D:1FC7    sub      ax,[bx+16AA]
1D8D:1FCB    xchg     al,ah

```

```

1D8D:1FCD      mov     al,7
1D8D:1FCF      sub     al,ah
1D8D:1FD1      xor     ah,ah
1D8D:1FD3      mov     cx,ax
1D8D:1FD5      lodsb
1D8D:1FD6      xor     al,dl
1D8D:1FD8      stosb
1D8D:1FD9      loop    1FD5
1D8D:1FDB      mov     [bx+16BE],di
1D8D:1FDF      xor     ax,ax
1D8D:1FE1      mov     ds,ax
1D8D:1FE3      ret

```

A PROLOK 2.0 kódja ezután már régi ismerősként köszön vissza:

```

1D8D:2850      mov     cs:[1F08],bx
1D8D:2855      mov     bx,cs:[1EB4]
1D8D:285A      xor     cs:[1EB6],bx
1D8D:285F      shl     bx,1
1D8D:2861      shl     bx,1
1D8D:2863      shl     bx,1
1D8D:2865      shl     bx,1
1D8D:2867      mov     cs:[bx+2722],ds
1D8D:286C      mov     cs:[bx+2724],es
1D8D:2871      mov     cs:[bx+2726],ax
1D8D:2876      mov     ax,cs:[1F08]
1D8D:287A      mov     cs:[bx+2728],ax
1D8D:287F      mov     cs:[bx+272A],cx
1D8D:2884      mov     cs:[bx+272C],dx
1D8D:2889      mov     cs:[bx+272E],si
1D8D:288E      mov     cs:[bx+2730],di
1D8D:2893      shr     bx,1
1D8D:2895      shr     bx,1
1D8D:2897      shr     bx,1
1D8D:2899      pop     cs:[bx+1EA4]
1D8D:289E      pop     cs:[1F08]
1D8D:28A3      pop     cs:[bx+1EAC]
1D8D:28A8      add     bx,+02
1D8D:28AB      cmp     bx,+08
1D8D:28AE      jc      28B2
1D8D:28B0      xor     bx,bx
1D8D:28B2      shr     bx,1
1D8D:28B4      mov     cs:[1EB4],bx
1D8D:28B9      sti
1D8D:28BA      push    cs
1D8D:28BB      pop     ds
1D8D:28BC      inc     w,[1EC4]
1D8D:28C0      pop     ds
1D8D:28C1      push    ds
1D8D:28C2      xor     ax,ax
1D8D:28C4      mov     es,ax
1D8D:28C6      mov     di,0
1D8D:28C9      mov     si,bp
1D8D:28CB      lodsw
1D8D:28CC      stosw
1D8D:28CD      mov     bx,cs

```

```

1D8D:28CF      mov      ax,bx
1D8D:28D1      stosw
1D8D:28D2      add      di,+04
1D8D:28D5      mov      cx,3
1D8D:28D8      test     w,cs:[7F2],3
1D8D:28DF      jnz      28E5
1D8D:28E1      add      di,+04
1D8D:28E4      dec      cx
1D8D:28E5      lodsb
1D8D:28E6      stosb
1D8D:28E7      lodsb
1D8D:28E8      stosb
1D8D:28E9      mov      ax,cs:[1EC4]
1D8D:28ED      and      ax,7
1D8D:28F0      add      bx,ax
1D8D:28F2      mov      es:[di],bx
1D8D:28F5      add      bx,cx
1D8D:28F7      add      di,+02
1D8D:28FA      loop     28E5
1D8D:28FC      push     cs
1D8D:28FD      pop      ds
1D8D:28FE      call     2987
1D8D:2901      call     29AD
1D8D:2904      cmp      w,[4],2850
1D8D:290A      jz       290E
1D8D:290C      cli
1D8D:290D      hlt
1D8D:290E      push     cs
1D8D:290F      pop      ds
1D8D:2910      mov      ax,[1EC4]
1D8D:2913      mov      cx,16
1D8D:2916      mov      bx,0FFFF
1D8D:2919      inc      bx
1D8D:291A      cmp      al,[bx+1EC6]
1D8D:291E      loopnz   2919
1D8D:2920      jnz      293E
1D8D:2922      push     ax
1D8D:2923      push     bx
1D8D:2924      push     cx
1D8D:2925      shl      bx,1
1D8D:2927      call     [bx+1EDC]
1D8D:292B      cmp      ax,0
1D8D:292E      jz       2934
1D8D:2930      call     ax
1D8D:2932      jmps     292B
1D8D:2934      pop      cx
1D8D:2935      pop      bx
1D8D:2936      pop      ax
1D8D:2937      push     cs
1D8D:2938      pop      ds
1D8D:2939      mov      b,[bx+1EC6],0
1D8D:293E      jcxz     2942
1D8D:2940      jmps     2919
1D8D:2942      xor      ax,ax
1D8D:2944      mov      es,ax
1D8D:2946      mov      w,es:[4],2850
1D8D:294D      mov      bx,[1EB4]

```



```

1D8D:2951    shl      bx,1
1D8D:2953    push     cs:[bx+1EAC]
1D8D:2958    push     cs
1D8D:2959    push     cs:[bx+1EA4]
1D8D:295E    mov      cl,3
1D8D:2960    shl      bx,cl
1D8D:2962    mov      es,[bx+2724]
1D8D:2966    mov      ax,[bx+2726]
1D8D:296A    mov      cx,[bx+2728]
1D8D:296E    mov      [1F08],cx
1D8D:2972    mov      cx,[bx+272A]
1D8D:2976    mov      dx,[bx+272C]
1D8D:297A    mov      si,[bx+272E]
1D8D:297E    mov      di,[bx+2730]
1D8D:2982    mov      ds,[bx+2722]
1D8D:2986    iret

```

A két ismerős szubrutin:

```

1D8D:2987    mov      bx,cs:[1EB6]
1D8D:298C    shl      bx,1
1D8D:298E    mov      di,[bx+1EA4]
1D8D:2992    sub      di,+07
1D8D:2995    mov      si,di
1D8D:2997    mov      ax,cs
1D8D:2999    mov      es,ax
1D8D:299B    mov      ds,ax
1D8D:299D    mov      cx,5
1D8D:29A0    lodsb
1D8D:29A1    xor      al,bl
1D8D:29A3    mov      al,bl
1D8D:29A5    add      bx,[1EC4]
1D8D:29A9    stosb
1D8D:29AA    loop     29A0
1D8D:29AC    ret

```

valamint

```

1D8D:29AD    push     cs
1D8D:29AE    pop      ds
1D8D:29AF    mov      bx,[1EB4]
1D8D:29B3    mov      dl,[bx+1EC0]
1D8D:29B7    shl      bx,1
1D8D:29B9    mov      di,[bx+1EB8]
1D8D:29BD    mov      si,di
1D8D:29BF    mov      ax,di
1D8D:29C1    push     cs
1D8D:29C2    pop      es
1D8D:29C3    sub      ax,[bx+1EA4]
1D8D:29C7    xchg     al,ah
1D8D:29C9    mov      al,7
1D8D:29CB    sub      al,ah
1D8D:29CD    xor      ah,ah
1D8D:29CF    mov      cx,ax
1D8D:29D1    lodsb
1D8D:29D2    xor      al,dl

```

1D8D:29D4	stosb	
1D8D:29D5	loop	29D1
1D8D:29D7	mov	[bx+1EB8],di
1D8D:29DB	xor	ax,ax
1D8D:29DD	mov	ds,ax
1D8D:29DF	ret	

Mindegyik verzió végén megtalálható az ominózus, sokaknak fejtörést okozó IRET, valamint a különböző fontos részek (kulcslemez-ellenőrzés, főprogram-konvertálás, stb.) hívása azonos módon történik: ahányszor ez a ciklus lefut, annyszor inkrementálódik a számláló. Ha ez azonos lesz egy táblázat valamely értékével, egy másik táblázatból veszi a program az AX értékét, konvertálja az AX címen levő rutint, és végrehajtja a CALL AX-et.

Az első CRYPTO változatok nem túl szerencsésen javították a PROLOK 1.07-es verziójának azt a hibáját, hogy gondatlanul bánt az INT 9-cel. Szorgalmasan levizsgálták az összes megszakítás vektort, és ha valahol szabálytalanságot vettek észre, egy barátságtalan "Nem IBM kompatibilis rendszer" hibaüzenettel visszaadták a vezérlést a DOS-nak. Így azok a programozók, akik a német (és persze a magyar) billentyűzet kiosztást, a SIDEKICK általános utility-t, illetve a külföldi fejlesztők körében elterjedt ATRON BUGBUSTER bővítőkartát használták, kétségbeejtő "információt" kaptak az esetleg eredeti IBM gépeikről is, míg rá nem jöttek, hogy kedvenc utility-jüket és a levédett szoftvert nem használhatják egyszerre. A későbbi CRYPTO-ból ez hiányzik, az INT 9 jól érvényesülhet. Mivel a lézerlyukat nem tudták reprodukálni, különleges track-formázást hajtottak végre, valamint a konverzió során a program bizonyos helyeiből egy konstanst (57-et) vontak le.

SUPERLOCK

A másik híres-hírhedt védelem a SOFTGUARD cég "SUPERLOCK" védelme. Mint már jeleztük, valójában nem is olyan "super": INT 9-et használó debuggerekkel jól nyomkövethető. Sőt a PROLOK új, 2.0-ás verziójától eltérően nem írja át a gép TIMER TICK (INT 1C) megszakítását, mely kapcsolatban van a belső órával. Mindezek ellenére a SUPERLOCK megfejtése nem triviális, az INT 9-et használó debuggerekkel talán egy kicsit nehezebb, mint a PROLOK megfejtése.

Először is miről lehet felismerni a SOFTGUARD védelmet? A SOFTGUARD védelem tulajdonképpen hasonló jellegű programok gyűjtőneve. Rengeteg verziója van és még nem is mindig védelem a szó megszokott értelmében: a dBASE-III plus szoftver többfelhasználós részére, a dBASE Administrator-ra SUPERLOCK van ráinstallálva, mégis másolható. A SUPERLOCK feladata itt nem a másolás, hanem a hozzáférés védelme: az Administrator eredeti alakjában tetszőleges számú felhasználót tud a hálózaton keresztül kiszolgálni. Az ASHTON TATE cég, a dBASE gyártója ezt "butított" változatban dobja a piacra: a SOFTGUARD feladata az, hogy ne engedje meg, hogy a szoftvert egy hálózatban egyszerre háromnál több felhasználó használja. Minden további három felhasználó hozzáférését a dBASE LAN PACK biztosítja annyszor, ahányszor azt megveszik.

A SOFTGUARD védelmet (vagy a LAN PACK-szerű "felhasználó szűrőt") mindig arról lehet felismerni, hogy a floppy lemezen vagy winchester meghajtón bizonyos CML????

.FCL, illetve VDF????VDF (jelen esetben a kérdőjelek számjegyeket helyettesítenek) nem látható (titkos) fájlok vannak. Ezeket a DISK EXPLORER-rel meg lehet tekinteni. (Mint tudjuk, azáltal lesz egy fájl láthatatlan, hogy a directory bejegyzésében a fájl attribútum ennek megfelelően van beállítva). Felismerhető a SUPERLOCK arról is, hogy a védett program a védelem installálása után mindig .COM kiterjesztésű, függetlenül attól, hogy eredetileg milyen volt. Az ilyen típusú védelem működése így több ponton is eltér a már ismertetett PROLOK-tól. Míg egy PROLOK program vázlatosan így néz ki:

védelem	program
---------	---------

és a védelem fejrésze ellenőrzi a kulcslemezt, valamint bonyolult módszerrel konvertálja a rögtön utána következő programot, a SOFTGUARD-védelem lefutása több, jól elkülönülő fázisra osztható:

- 1) A védelem egy része betöltődik, majd áthelyezi magát a memóriának számára elérhető legfelső részébe, épp hogy a DOS ott levő részét ne bántsa.
- 2) A védelem FCL, VDF, stb. fájlokban levő részei is betöltődnek, szintén a memóriának számukra elérhető legmagasabb részébe, majd több konverzió, INT 1 és INT 3 felülírása stb. után lemez ellenőrzés történik (ez lehet winchester ellenőrzés, floppy ellenőrzés, sőt, mint láttuk, el is maradhat).
- 3) Ha a védelem rendben talál mindent, elkezdődik a lemezen levő (sokszor láthatatlan) fő fájlok betöltése. Ez az a fájl, amely valamikor maga a program volt, most azonban a felismerhetetlenségig konvertálva van.
- 4) A betöltés után megkezdődik a visszakonvertálás, és az így visszakonvertált program lesz végrehajtva.

PROLOK védelem esetén a hasonló konvertálás után az eredeti program rövid ideig úgy néz ki, mint a PROLOK installálása előtti, a gyártó által kifejlesztett példány. Mivel a konvertáló rutin viszonylag hosszabb ideig fut le (1-2 másodperc), gyors billentyű leütéssel (INT 9 megszakítás!) pontosan a rutin közepén találjuk megunkat. A konvertáló rutinban nincs INT 1 és INT 3 felülírás illetve használat (milyen végzetes hiba!), így azt debugger alatt lefuttatva egyszerűen kimenthetjük a memória tartalmát. A SUPERLOCK első verziói elkövezték szóról szóra ugyanezt a hibát, így a megfejtés gyors reflexszel legfeljebb öt percet vett igénybe. Később egy kicsit módosították a módszert: az eredeti program bizonyos fontosabb adatait (mint tényleges hossz, .EXE fájlokra jellemző "MZ" kezdet, stb.) máshol, illetve egyáltalán nem tárolták, de mindezt nem a gyors reflexek elleni védekezéséért, hanem a már leírt UNGUARD és NOGUARD megtévesztésére. A "kézi" módszerrel így már akár egy órát is el lehet bíbelődni, ha nincs szerencsénk. A végén azonban biztos a siker.

Az igazán új, amit a SUPERLOCK hozott az, hogy a védelem már nem szükségszerűen floppyhoz kötött, a védett programhoz jár többnyire olyan winchesterre installáló utility, mely csak korlátozott számú installálást enged meg ugyan, de legalább nem kell a "kulcslemezt" magunknál tartani. Mivel winchesterre különleges formázást nem szokás írni, itt a különböző fájlok abszolút és egymáshoz képesti, relatív elhelyezkedését ellenőrzi a védelem, s másik winchester meghajtón a védelem nem fog futni.

(Itt elkövették az uniformizálás hibáját: a SOFTGUARD-védelem más, csak nagyon kevés IBM-kompatibilis gépeken is létezik, mint pl. az APRICOT-PC. Olyan kódot használtak ennél a védelemnél, mely gyakorlatilag ugyanúgy fut az IBM-PC-n és az APRICOT-on. Egy APRICOT-ra installált APRICOT-dBASE összes, így "titkos" fájlját is az APRICOT winchesteréről soros vonalon az IBM XT winchesterére átküldve, az ugyan először nem fog elindulni, de egyszerű olyan programot írni, mely a TIMER TICK-en, azaz INT 1C megszakításon keresztül a leírt módon/ mintegy felügyel a védelem futására, és kellő időben azt átírja. Az ilyen módon "leszimulált" eredeti winchesteren a védelem mindent rendben talál és az UNGUARD, mely egyébként az APRICOT-on nem fut, le is szedi a SUPERLOCK-ot. Hasonló sikerre vezet az INT 9-es "kézi" módszer. APRICOT esetén a SOFTGUARD annyira biztos volt magában, hogy nem vette a fáradságot, hogy a védelmükre egyébként jellemző, az előbbiekben leírt módszerrel az .EXE adatokat elrejtse !)

Összefoglalva a SUPERLOCK adta új lehetőségeket, azt állapíthatjuk meg, hogy:

- a SUPERLOCK winchesterre installálható, hálózaton is üzemeltethető védelem.
- a SUPERLOCK másolóprogramokkal nem másolható, ugyanis a lemez néhány track-jét nem a közönséges PC-kontrollerrel írták fel.
- a SUPERLOCK installálása winchesterre eléggé nehézkes, a laikus tévedhet, és esetleg elronthatja.
- a SUPERLOCK-ot általában a NOGUARD vagy UNGUARD meg tudja kerülni és "visszainstallálja", felszabadítva ezáltal az eredeti programot.
- bár a SUPERLOCK kódja kellően átláthatatlan és kellően átjárhatatlan a közönséges debugger-ek számára, az új eszközök könnyen megbirkóznak vele.

Példaként lássuk a SUPERLOCK fájlkonvertáló rutinját, INT11-gyel való "kellemetlenkedését", és végül a konvertált fájl elejét!

```

99F3:0108      mov     bx,[bp+0C]
99F3:010B      mov     es,bx
99F3:010D      mov     bx,[bp+0A]
99F3:0110      call    190
99F3:0113      mov     dx,[0C]
99F3:0117      mov     b,[18],11
99F3:011C      mov     di,0
99F3:011F      mov     cx,0
99F3:0122      cmp     dx,80
99F3:0126      jbe     12B
99F3:0128      mov     dx,80
99F3:012B      sub     [0C],dx
99F3:012F      mov     al,es:[bx]
99F3:0132      mov     ah,al
99F3:0134      mov     cl,[18]
99F3:0138      xor     cx,di
99F3:013A      sub     al,cl
99F3:013C      mov     [18],ah
99F3:0140      mov     si,di
99F3:0142      and     si,7
99F3:0146      mov     cx,si

```

```

99F3:0148      add      cl,[si+4]
99F3:014C      mov      si,cx
99F3:014E      and      si,7
99F3:0152      add      al,[si+4]
99F3:0156      mov      cl,[si+4]
99F3:015A      and      cl,7
99F3:015D      rol      al,cl
99F3:015F      mov      si,di
99F3:0161      and      si,7
99F3:0165      mov      cl,[si+4]
99F3:0169      xor      al,cl
99F3:016B      mov      es:[bx],al
99F3:016E      inc      bx
99F3:016F      inc      di
99F3:0170      dec      dx
99F3:0171      jnz      12F
99F3:0173      cmp      w,[0C],+00
99F3:0178      jnz      113
99F3:017A      mov      al,0
99F3:017C      jmps     17F
99F3:017E      nop
99F3:017F      mov      ah,0
99F3:0181      pop      dx
99F3:0182      pop      cx
99F3:0183      pop      bx
99F3:0184      pop      di
99F3:0185      pop      si
99F3:0186      pop      es
99F3:0187      pop      ds
99F3:0188      pop      bp
99F3:0189      cld
99F3:018A      cmp      al,0
99F3:018C      jbe      18F
99F3:018E      stc
99F3:018F      ret      L
99F3:0190      push     ax
99F3:0191      push     cx
99F3:0192      push     di
99F3:0193      mov      ax,bx
99F3:0195      mov      cl,4
99F3:0197      shr      ax,cl
99F3:0199      mov      di,es
99F3:019B      add      di,ax
99F3:019D      mov      es,di
99F3:019F      and      bx,0F
99F3:01A3      pop      di
99F3:01A4      pop      cx
99F3:01A5      pop      ax
99F3:01A6      ret

```

A SUPERLOCK-nak nincs a PROLOK-hoz hasonló főciklusa, helyette a program "lineárisan" fut le, különböző CALL-okat hajtva végre. Mindez azonban ugyanolyan nehezen követhető, mint a PROLOK: a CALL-ok belsejében újabb és újabb CALL-ok tűnnek fel, INT1 és INT3 hívásokkal kombinálva. Az itt látható rutin többször is

meghívódik a futás során, és ha a 0:4-en, 0:C-n és 0:8C-n levő címek nem stimmelnek, végtelen ciklusba kerül a program.

```

995F:043C      mov     ax,0
995F:043F      push    es
995F:0440      mov     es,ax
995F:0442      cmp     es:[4],ax
995F:0447      jz      44E
995F:0449      call    570
995F:044C      jmps    442
995F:044E      lea     bx,[280]
995F:0452      cmp     es:[8C],bx
995F:0457      jz      45E
995F:0459      call    5F0
995F:045C      jmps    44E
995F:045E      cmp     es:[0C],ax
995F:0463      jz      46A
995F:0465      call    670
995F:0468      jmps    45E
995F:046A      nop
995F:046B      pop     es
995F:046C      mov     b,[3D],3
995F:0471      lea     ax,[43C]
995F:0475      mov     [49],ax
995F:0478      mov     [4B],cs
995F:047C      lea     cx,[49E]
995F:0480      sub     cx,ax
995F:0482      mov     [47],cx
995F:0486      mov     al,3

```

A továbbiakban láthatjuk az éppen visszakonvertálódó BMDP/PC indítóprogramjának az elejét. (A közkedvelt BMDP statisztikai programcsomag a SUPERLOCK egy korai verziójával van levédve.)

ax=2903 bx=004D ds=995F es=97FB cs=995F ss=9A0E bp=0E48
ds:[003F]=03

```

1D8C:0000      4D 5A BD 00 34 00 0E 00-20 00 41 00 FF FF 4C 06
1D8C:0010      00 04 3E 91 0B 00 00 00-1E 00 00 00 01 00 09 00
1D8C:0020      00 00 2C 00 00 00 00 F0 00-00 00 7D 10 00 00 88 10
1D8C:0030      00 00 21 13 00 00 74 0A-A2 04 30 29 00 00 45 2A
1D8C:0040      00 00 B8 40 00 00 E1 42-00 00 52 43 00 00 7C 43
1D8C:0050      00 00 53 44 00 00 00 00-00 00 00 00 00 00 00 00
1D8C:0060      00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
1D8C:0070      00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
1D8C:0080      00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
1D8C:0090      00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
1D8C:00A0      00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
1D8C:00B0      00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
1D8C:00C0      00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
1D8C:00D0      00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
1D8C:00E0      00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
1D8C:00F0      00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
1D8C:0100      00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00

```


1D8C:0110	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
1D8C:0120	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
1D8C:0130	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
1D8C:0140	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00

Kénytelenek vagyunk azt mondani, hogy ahogyan a PROLOK és magyar utánzatai, úgy a SUPERLOCK sem tekinthető már nagyon jó védelemnek, bár kezdetben ezek a védelmek eléggé hatásosak voltak.

A SUPERLOCK-nak köszönheti létrejöttét egy új másolóeszköz, a COPY II OPTION BOARD nevű bővítőkártya. A COPY II PC másolóprogram létrehozói nem tudtak kibé-
különni azzal, hogy "csak" a NOGUARD segítségével lehet a SUPERLOCK-ot másolni,
pontosabban eltávolítani, a közönséges másolóprogramok képtelenek az eredeti SOFT-
GUARD lemezeket másolni, mivel a védelem a PC kontrollerétől eltérő kontrollert hasz-
nál felírásra. Az OPTION BOARD egy olyan (az Egyesült Államokban 70-95 dollár,
Európában kb. 350 DM áron kapható) bővítőkártya, mely teljes egészében kiváltja a kö-
zönséges kontrollert és sok olyan felírást képes létrehozni, amit az nem. A COPY
BOARD megjelenésével érezhető törekvés kezdődött arra, hogy egyrészt az eddig védett
programokat a gyártók már védelem nélkül árulják (dBASE, PARADOX, FRAME-
WORK), másrészt olyan védelmeket kezdenek kifejleszteni, melyek a COPY BOARD-nak
is ellenállnak, illetve nem floppy-centrikusak.

Az utóbbiakra példák a különböző csatlakozásokra ráhelyezendő kulcs-kártyák (pl.
NOVELL hálózati szoftver), ill. fekete dobozok (AUTOCAD 2.5) stb... Ezek hatékonysá-
ga máris megkérdőjelezhető, ugyanis terjednek már idehaza az ezektől a hardveres védel-
meiktől megszabadított kalózmásolatok.

Az előbbieket kapcsán várható a PROLOK másodvirágzása, valamint a nem "lyukas" védel-
mek terjedése. Az ilyen nem "lyukas" védelemre, melynek lemez formázását a COPY
BOARD nem képes másolni, példa az "EVERLOCK" (Az-Tech cégtől), vagy a már tár-
gyalt SUPERLOK 3.0-ás verziója. Ez az EVERLOCK egy furcsa termék: ránézésre ideá-
lis. Nincs esélye itt sem a NOGUARD-nak és társainak, sem a COPY BOARD-nak, sem a
COPYWRIT-nak. Az EVERLOCK winchesterre installálható, a winchester ezután el-
menthető streamer-re és vissza, s a program közben nem vész el. (A SUPERLOCK ezt
nem engedi meg!) Ha egy kicsit jobban megnézzük az EVERLOCK-ot, láthatjuk, hogy
egyrészt a kód nem kellően szövevényes, a debug viszonylag könnyen megbirkózik vele
(nem egy kategória a PROLOK-kal vagy a SOFTGUARD-dal!), de "természetesen" az
INT 9 végzetes csapdáját sem kerüli el. A megfejtés és a védelem eltávolítása így hagyomá-
nyos eszközzel a szakember számára legfeljebb egy órai időt vesz igénybe, de inkább keve-
sebbet, a "kézi" módszerrel az amatőr megfejtő számára is legfeljebb 5-10 perc.

1B4F:0A1A	push	ax
1B4F:0A1B	mov	ax,0
1B4F:0A1E	mov	es,ax
1B4F:0A20	pop	ax
1B4F:0A21	lea	bx,[3D2]
1B4F:0A25	mov	es:[0C],bx
1B4F:0A2A	mov	es:[0E],cs
1B4F:0A2F	pop	dx
1B4F:0A30	pop	cx

1B4F:0A31	pop	bx
1B4F:0A32	pop	di
1B4F:0A33	pop	si
1B4F:0A34	pop	es
1B4F:0A35	ret	

A fenti EVERLOCK-részlettel szemben (ennyi az egész "kellemetlenkedés" INT1 és INT3 ügyben, pontosabban INT3 ügyben az egész védelem során!) térjünk vissza egy pillanatra a PROLOK 1.07-hez! Az itt látható rutin még a már ismertetett ciklus előtt van, a ciklus csak valamivel később kezd el működni.

1D86:17CC	cli	
1D86:17CD	mov	di,400
1D86:17D0	mov	cs:[1894],ss
1D86:17D5	mov	cs:[1896],sp
1D86:17DA	xor	ax,ax
1D86:17DC	mov	es,ax
1D86:17DE	mov	ax,es:[8]
1D86:17E2	mov	cs:[18A7],ax
1D86:17E6	mov	ax,es:[0A]
1D86:17EA	mov	cs:[18A9],ax
1D86:17EE	push	cs
1D86:17EF	pop	es
1D86:17F0	mov	ax,0F800
1D86:17F3	mov	ss,ax
1D86:17F5	mov	sp,8014
1D86:17F8	mov	ax,20CC
1D86:17FB	mov	dx,2F31
1D86:17FE	push	cs
1D86:17FF	mov	bx,0B89
1D86:1802	add	bx,cs:[1898]
1D86:1807	push	ax
1D86:1808	stc	
1D86:1809	sbb	ax,bx
1D86:180B	push	bx
1D86:180C	mov	cl,cs:[189A]
1D86:1811	sar	ax,cl
1D86:1813	push	ax
1D86:1814	pop	ax
1D86:1815	mov	cx,cs:[189B]
1D86:181A	mov	si,40
1D86:181D	shr	cx,1
1D86:181F	inc	cx
1D86:1820	lodsw	
1D86:1821	push	ax
1D86:1822	loop	1820
1D86:1824	mov	cx,55
1D86:1827	add	sp,+08
1D86:182A	push	cs
1D86:182B	push	dx
1D86:182C	add	sp,+0C
1D86:182F	push	ss
1D86:1830	pop	es
1D86:1831	add	sp,+78
1D86:1834	sub	sp,214
1D86:1838	mov	ax,cs:[1894]

```

1D86:183C    and        cx,7
1D86:1840    stc
1D86:1841    mov        ss,ax
1D86:1843    mov        bx,cs:[1896]
1D86:1848    mov        sp,bx
1D86:184A    sti
1D86:184B    rol        di,cl
1D86:184D    xor        di,10
1D86:1851    sub        w,es:[di],1B2
1D86:1856    push       es
1D86:1857    xor        ax,ax
1D86:1859    mov        es,ax
1D86:185B    mov        ax,cs:[18A7]
1D86:185F    mov        es:[8],ax
1D86:1863    mov        ax,cs:[18A9]
1D86:1867    mov        es:[0A],ax
1D86:186B    pop        es
1D86:186C    push       cs
1D86:186D    pop        ds
1D86:186E    cll
1D86:186F    xor        bx,bx
1D86:1871    mov        cx,18
1D86:1874    mov        si,189D
1D86:1877    sub        w,es:[8004],10E1
1D86:187E    lodsw
1D86:187F    sub        w,es:[di],0C8
1D86:1884    add        bx,ax
1D86:1886    into
1D86:1887    loop       187E
1D86:1889    xor        bx,1000
1D86:188D    jns        186F

```

Kiemelendő itt egyrészt a STACK ráhelyezése az interrupt táblára (de milyen ügyesen: F800:8014 lesz az SS:SP, azaz

$$\begin{array}{r}
 \text{F8000} \\
 + \quad 8014 \\
 \hline
 00014
 \end{array}$$

tehát SS:SP = 0:14 !), másrészt a 187E-s hurokban az INTO nagyon kellemetlen, és ez még csak a PROLOCK eleje. E példából is látható, hogy az 1986-os EVERLOCK születése pillanatában már túlhaladott volt: elég ránézni az 1984-es, többször is megfejtett PROLOCK-ra.

A SUPERLOCK 3.0 számunkra tulajdonképpen érdektelen: azon kívül, hogy a COPY II BOARD régi meghajtó programjai nem másolják, újdonságot nem tartalmaz, "kézi" módszerrel ez a védelem is fél óra alatt megfejtethető.

A legtöbb, védelmet gyártó cég felismerte azt, hogy a felhasználók egyrészt "nem szeretik" a kulcslemezt, mivel az esetleg megsérülhet, a program kulcslemezzel való indítása mindig körülményesebb és lassúbb, mint a merevlemezzel való "sima" indítás, másrészt a hálózatok terjedése is abba az irányba hatott, hogy létrejöttek a merevlemez védelmek.

A winchesteres védelem hátránya, hogy a winchestert olvasni csak a DOS-on keresztül szabad, ellentétben a floppy-val, melyet minden normálisabb védelem közvetlenül I/O portokat címezve (ír/olvas. Három ismertebb védelem létezik:

- 1.) Rejtett fájlok létének vizsgálata. Itt az a baj, hogy
 - a speciális programok (pl. PC-TOOLS) egyszerűen másolják ezeket is;
 - ha már felfedeztük őket, átírhatjuk "normál"-lá (pl. EXPLORER-rel, de alkalmas erre az ALTER parancs is), majd COPY-val másolható a szoftver, és a felhasználás helyén ismét rejtetté tehető.
- 2.) Egy fájl pontos helyének vizsgálata a winchesteren (pl. SUPERLOCK). A probléma az, hogy INT 25 rutint kénytelen használni a program.
- 3.) A winchester legelején levő partícióábrába beírás (pl. Vault cég HDLOCK-ja). Itt az a probléma, hogy
 - nem minden winchesteren működik jól;
 - korlátozott számú program installálható a winchesterre (csak 4-6 darab!)
 - felfedezve ezt, igen könnyű a felhasználandó winchesteren ugyanezt bevinni a winchester legelső szektorába.

Joggal feltehetjük a kérdést, hogyan lehetséges, hogy több ember heteken, hónapokon át kifejleszt egy igen áttekinthetetlen kódú védőprogramot, s ezek a programok mégis mind figyelmen kívül hagyják az INT 9 megszakítást és sokszor az INT 1C-t is (nem is beszélve az INT 2-ről, a hardver-megszakításról, azaz más néven NMI-ről). A CRYPTO első verzióiban érezhető volt a törekvés, hogy a PROLOK eme hibáját kijavítsák, de a próbálkozás nagyon szerencsétlenül sült el.

A teljesség kedvéért hozzátesszük, hogy van erre ellenpélda is: a magyar ELTSOFT cég EltGUARD floppys védelme az egész megszakítás-táblát először egy külön erre a célra szolgáló munkaterületre kimentti, utána felülírja arra az időre, amíg az ellenőrzés és a konvertálás tart, majd a legvégén a munkaterületről visszaírja az eredeti értékeket. Az INT 9-es "kézi" módszer így hatástalan, és a megfejtőnek nem marad más választása, mint hagyományos módszerrel nékilátni a PROLOK és SOFTGUARD bonyolultságát, szövevényességét is felülmúló kód megfejtésének. A másolat készítését meggátolja az is, hogy az ELTGUARD 3.2-es verzióját a COPY II BOARD meghajtó programjai sem képesek át-másolni.

"Lapzárta után" érkezett meg hozzánk a kulcskártyás védelem legreprezentatívabb képviselője, a NOVELL NetWare (ADVANCED NETWARE/86 ill. /286, 2.0A verziószámmal). A kulcskártyáról előzőleg a legkülönbözőbb rémhírek keringtek, az azt övező misztika felülmúlta a "lézeres" lemezek körüli misztikát is. Azonban a védelem nagyon-nagyon gyengére sikerült! A kulcskártyán levő sorozatszám kiolvasását egyetlen rutin végzi, bizonyos konvertálás után ennek a programban levő sorozatszámmal kell egyeznie. Az egyeztetést is egyetlen(!), jól látható és debuggolható(!) rutin hajtja végre. Így ez a kulcskártyás védelem a leggyengébb védelmek közé sorolható.

A tömegáruként forgalmazott programok esetén nem lehet egyedi installációról beszélni, a mikrogépeknek pedig nincs saját processzor azonosító kódjuk, amely a védelmet támogatná.

A programok védelmére ugyanakkor nagy szükség van, mivel a jó szoftver ára sok esetben nem alacsony, a másolást engedélyezők pedig mentesítve érzik magukat azáltal, hogy ők csak egy-egy személyes ismerősüknek adták át szívésségből a programot, szó sincs tehát haszonszerzésről vagy visszaélésről.

III. ENCRYPTION védelem

(Siffrírozás, jelszó védelem)

Egészen más feladatokat lát el a siffrírozás (ENCRYPTION), mint a másolásvédelem. A siffrírozásnak ott van szerepe, ahol különböző felhasználók osztozhatnak ugyanazon a mágneses tárolón (ez többnyire winchester, de olykor floppy lemez is lehet), és nem kívánatos, hogy a különböző információkhoz, programokhoz az arra illetéktelen személy is hozzájusson. Míg a saját processzor-azonosító hiánya a mikrogépeknél merőben más helyzetet eredményez a programvédelem területén, mint a nagy számítógépeknél, a siffrírozás esetén körülbelül hasonló elvek szerint történik minden: adott egy siffrírozó-dekódoló program, azt előhívja a felhasználó, megmondja neki, hogy melyik állományt szeretné titkosítani, a siffrírozó program megkérdezi a jelszót, és a jelszó alapján különböző, bonyolult algoritmusok segítségével a régi állományból létrehoz egy újat, mely siffrírozva van. Ezzel párhuzamosan (opcionálisan) törli a könyvtárból a régi állományt (és esetleg a helyét is felülírja, mivel ha csak a directory bejegyzést törli, utána az eredeti állomány visszaállítható lenne.)

Különböző komfortfokozatú és algoritmus-bonyolultságú siffrírozó programok vannak a piacon, egy nem teljes felsorolásuk:

LOCKSMITH/PC (pontosabban annak ilyen opciója);
SUPERKEY (annak szintén opciója), s egy európai termék, az
MPROTECT.

Az MPROTECT bejelentését nagy hírveréssel beharangozott versennyel kötötték össze: az első kétszáz érdeklődőnek, aki elküldte címét a gyártó cégnek, visszaküldtek egy demonstrációs verziót, rajta egy siffrírozott állománnyal. Megadták még a kód első három karakterét is, és kihirdették, hogy a kód első három megfejtője egy kb. 20000 DM (egy új Mercedes gépkocsi ára!) értékű számítógépet kap, nevezetesen COMPAQ 386-ot.

A következő oldalakon látható, hogy a szoftvergyártó az értékes díj kitűzésével nem vállalt túl nagy kockázatot: matematikailag nagyobb az esélyünk arra, hogy egy mindig azonosképpen kitöltött LOTTO szelvénnel a világ három országát egy hónapon belül beutazva mindenütt megnyerjük a főnyereményt, mint a kódot kitalálni.

Az MPROTECT, mint az összes többi hasonló termék, nem mentes a hibáktól:

- működése lassú, a rövidebb fájlok konvertálása is időigényes, s ezt a manuális munka még lassítja;
- munka közben az eredeti adatok vannak a lemezen, ami pl. hálózatban megengedhetetlen;
- nincs megoldva a hierarchikus védelem, azaz az "amit szabad Jupiternek, azt nem szabad az ökörnek" elv;
- a védelem (és jelszó) nem személyhez, hanem fájlokhoz kötött;
- egy alkalmazás sokszor több tucat fájlból áll. Mindegyiket oda-vissza siffrírozni fáradtságos, hibákhoz vezet (ugyanis a konvertálást minden egyes fájlra kézzel kell végrehajtani, munka előtt és után is!).

Az utóbbi hibát néhány adatbáziskezelő szoftver is megpróbálja orvosolni, többnyire úgy, hogy egyszerűen a jelszó ismeretéhez köti az adatokhoz való hozzáférést (pl. KNOWLEDGEMAN, KNOWLEDGEMAN/2, DATAEASE, R:BASE, stb.), azonban gyakori hiba az, hogy a jelszó egy az egyben megtalálható valahol a lemezen, és az adatfájlok nincsenek siffrírozva. Hosszú ideig hiányzott a piacról egy olyan programtermék, mely memóriarezidens, és minden adatot önmagán, mint szűrőn keresztül átengedve oda-vissza automatikusan konvertál, s a hierarchikus hozzáférést is támogatja. Ma már ez elérhető, az EltSECRET szoftvert a következő fejezetben ismertetjük.

Nézzük meg az MPROTECT listáit!

Ez a rutin végzi a fájl kódolását. Látszólag a helyzet egyszerű: a fájlból a soron következő bájtot a BL-be kerül, valahogy képződik az AL, az levonódik a BL-ből, és az eredmény visszaíródik a fájlba:

2D6F:3B31	51	push	cx
2D6F:3B32	A0A728	mov	al,[28A7]
2D6F:3B35	A2D028	mov	[28D0],al
2D6F:3B38	8A1C	mov	bl,[si]
2D6F:3B3A	E80D00	call	3B4A
2D6F:3B3D	2AD8	sub	bl,al
2D6F:3B3F	881C	mov	[si],bl
2D6F:3B41	881EA728	mov	[28A7],bl
2D6F:3B45	46	inc	si
2D6F:3B46	E2EA	loop	3B32
2D6F:3B48	59	pop	cx
2D6F:3B49	C3	ret	

A probléma, mint rögtön látni fogjuk, éppen az AL képződése körül van, ennek újabb értékei beépülnek a kulcsszóba, a 28D0-s címre. A kulcsszó mindig hex. 20, azaz dec. 32 hosszú lesz, ha mi rövidebbet adunk meg, az szóközökkel kiegészül, illetve 7 darab szóköz (hex. 20) és blank (00) az általunk megadott szó elé kerül.

2D6F:3B4A	53	push	bx
2D6F:3B4B	51	push	cx
2D6F:3B4C	52	push	dx
2D6F:3B4D	56	push	si
2D6F:3B4E	53	push	bx
2D6F:3B4F	51	push	cx
2D6F:3B50	BBC928	mov	bx,28C9
2D6F:3B53	B91F00	mov	cx,1F
2D6F:3B56	51	push	cx
2D6F:3B57	8B07	mov	ax,[bx]
2D6F:3B59	B9F307	mov	cx,7F3
2D6F:3B5C	F7E1	mul	cx
2D6F:3B5E	051936	add	ax,3619
2D6F:3B61	8907	mov	[bx],ax
2D6F:3B63	59	pop	cx
2D6F:3B64	43	inc	bx
2D6F:3B65	E2EF	loop	3B56
2D6F:3B67	59	pop	cx
2D6F:3B68	5B	pop	bx

2D6F:3B69	BBC928	mov	bx,28C9
2D6F:3B6C	B92000	mov	cx,20
2D6F:3B6F	B600	mov	dh,0
2D6F:3B71	B80000	mov	ax,0
2D6F:3B74	8A17	mov	dl,[bx]
2D6F:3B76	03C2	add	ax,dx
2D6F:3B78	43	inc	bx
2D6F:3B79	E2F9	loop	3B74
2D6F:3B7B	B400	mov	ah,0
2D6F:3B7D	5E	pop	si
2D6F:3B7E	5A	pop	dx
2D6F:3B7F	59	pop	cx
2D6F:3B80	5B	pop	bx
2D6F:3B81	C3	ret	

Az alábbi programrészletben kulcsszónak XX-et adtuk meg:

ax=0200 bx=28C9 ds=2D6F es=2D6F cs=2D6F ss=2D6F bp=0936
ds:[28C9]=2020

2D6F:3B4E	53	push	bx
2D6F:3B4F	51	push	cx
2D6F:3B50	BBC928	mov	bx,28C9
2D6F:3B53	B91F00	mov	cx,1F
2D6F:3B56	51	push	cx
2D6F:3B57	8B07	mov	ax,[bx]
2D6F:3B59	B9F307	mov	cx,7F3
2D6F:3B5C	F7E1	mul	cx
2D6F:3B5E	051936	add	ax,3619
2D6F:3B61	8907	mov	[bx],ax
2D6F:3B63	59	pop	cx
2D6F:3B64	43	inc	bx
2D6F:3B65	E2EF	loop	3B56
2D6F:3B67	59	pop	cx
2D6F:3B68	5B	pop	bx
2D6F:3B69	BBC928	mov	bx,28C9
2D6F:3B6C	B92000	mov	cx,20
2D6F:3B6F	B600	mov	dh,0

2D6F:28C0	20 20 20 20 20 20 20 20	20-00	20 20 20 20 20 20 20 20
2D6F:28D0	00 78 78 20 20 20 20 20	20-20	20 20 20 20 20 20 20 20

És íme a kulcsszó környékéről való memória-dump, már az előző oldalról való, 2D6F:3B46-on levő LOOP 3B32 ciklus lefutása után:

2D6F:28C0	20 20 20 20 20 20 20 20	20-00	79 95 C3 F2 50 A8 9C
2D6F:28D0	9A 1D 4F 79 95 C3 F2	50-A8	9C BA 9D E9 A4 E0 4F
2D6F:28E0	79 95 C3 F2 50 A8 9C	BA-6C	00 78 78 20 20 20 20
2D6F:28F0	20 20 20 20 20 20 20 20	20-20	20 20 20 20 20 20 20 20
2D6F:2900	20 20 00 00 00 58 00	58-00	58 00 58 00 00 00 00
2D6F:2910	00 00 00 00 00 00 00 00	00-00	5F 2D 00 00 00 08 00
2D6F:2920	0F 0F 0F 0F 0F 0F 0F 0F	0F-00	00 00 00 00 00 00 00
2D6F:2930	8A 05 3A 04 75 08 47	46-E2	F6 F8 EB 02 90 F9 C3
2D6F:2940	EB 40 90 51 57 BF B0	28-80	3D 00 74 06 C6 05 20
2D6F:2950	47 BE F5 BF EA 28 80	3D-00	74 06 C6 05 20 47 EB

```

2D6F:2960  F5 5F 59 C3 B8 00 00 B6-00 8A 15 03 C2 47 E2 F9
2D6F:2970  3B C3 75 01 C3 E8 01 00-C3 BF 00 00 C6 05 00 47
2D6F:2980  EB FA B9 08 00 BF C7 0E-8B 1E 51 28 E8 D5 FF B9
2D6F:2990  08 00 BF BF 0E 8B 1E 93-0E E8 C8 FF B9 0C 00 BF
2D6F:29A0  D3 0E 8B 1E B1 0E B9 4E-00 BF B7 05 8B 1E A2 28
2D6F:29B0  E8 B1 FF C3 50 53 51 52-B4 2C CD 21 52 CD 21 52
2D6F:29C0  CD 21 59 5B 3B D9 74 07-3B CA 74 03 E8 AA FF 5A
2D6F:29D0  59 5B 58 C3 C7 06 86 28-01 00 C7 06 30 09 0B 00
2D6F:29E0  B8 02 00 BB 01 2A 2E A3-32 09 B8 32 09 2E 89 47
2D6F:29F0  FE B8 50 00 2E 89 47 FC-EB 07 90 00 00 50 00 32
2D6F:2A00  09 EB 03 90 46 00 BE 04-2A 8B 16 32 09 A1 30 09
    
```

Látható, hogy az előző, kezdeti állapothoz képest a 2D6F:28C9-ben elhelyezkedő kulcsszó teljesen megváltozott. A kiinduló állapotot azonban a program nem felejtette el: 2D6F:28E9-től láthatjuk azt a dump-ban, és amikor a következő fájlselet konvertálásakor megint ráfut a program a 2D6F:3B31-en kezdődő ciklusra, újra ott fogjuk azt találni.

Mivel a fent látott algoritmus és a kód első három karakterének ismeretében legfeljebb a kulcsszó első három karaktere derül ki, a többire $32-8-3=21$ "tipp hely" marad, ami szűkebb ASCII készlet esetén is $127-32=95$ eset tipp helyenként, és ez összesen $95^{21} > 64^{21} > 2^{8 \cdot 21} > 2^{160} > 1000^{16} > 10^{48}$ eset, azaz legalább 48 jegyű szám! A LOTTO-n az összes lehetséges eset száma $< 10^{10}$, három országban való nyerés valószínűsége $> 1:10^{30}$.

Nagyon egyszerűen be lehet látni, hogy a feladat egy nem lineáris, 24 ismeretlenes diofantoszi egyenlettel azonos, így ha találgatás helyett matematikai módszerhez folyamodnak, ezt az egyenletet kell megoldani. A probléma nehézségét illusztrálандó, megjegyezzük, hogy a sokkal egyszerűbb formájú $x^n + y^n = z^n$ háromismeretlenes diofantoszi egyenlet teljes diszkussziója minden természetes n számra (a híres Fermat-sejtés) még mindig megoldatlan matematikai probléma.

IV. Adatvédelmi rendszerek

IBM PC-n hosszú ideig nem volt szükség adatvédelemre. A UNIX alapú rendszerek ezen a gépkategórián nem terjedtek el. Az első, adathozzáférést szabályozó rendszerek a hálózati operációs rendszerek voltak.

LAN

A PC Network Program hozzáférés-szabályozása az erőforrás nevek szerint történik. Minden erőforrásnak (adatfájl, program, nyomtató) név adható, a hozzáférés ellenőrzésére ezekhez jelszavak köthetők. A probléma ebben a megoldásban az, hogy egyrészt a felhasználónak nem kell a rendszerbe bejelentkeznie (logonolnia) ahhoz, hogy ezeket elérje, másrészt igen sok jelszót kell ismernie munkájához. Ugyanígy dolgozik a 3COM 3+ hálózati rendszere is.

A Novell hálózat alapvetően a Server (központi gép) adatállományaihoz való hozzáférést szabályozza. Ez a szabályozás kizárólag alkönyvtár szintű, azaz azt lehet megadni, az egyes felhasználók mely alkönyvtárakhoz férhetnek hozzá, s ott milyen jogokkal rendelkeznek. A jogkörök széles skáláját nyújtja a Novell: keresési, olvasási, írási, létrehozási, törlési, átnevezési, módosítási jog. Az egyes jogok alkönyvtáranként és felhasználónként változhatnak. A felhasználók – mint szinte minden adatvédelmi rendszerben – névvel és jelszóval jelentkeznek be. A Novell – okulva más rendszerek hibáján – nemcsak védi a jelszavakat (kódolva tárolja őket a serveren), de az ellenőrzés sem a munkaállomáson történik. Ennek az a jelentősége, hogy nem lehet nyomkövetéssel sem megtudni a jelszavakat. (A munkaállomáson futó hálózati program a felhasználó által beírt jelszót átkódolja, majd elküldi a serveren futó programnak. Az igazolja vissza a jelszó helyességét vagy érvénytelenségét, így az ellenőrzésre a felhasználónak saját munkaállomásán nincs ráhatása.) Egyedi fájlok védelmére csak korlátozottan képes a NetWare: egy fájl tehető csak olvashatóvá (Read-Only), ill. kizárólagossá (Private), ezek azonban nem szabályozhatók felhasználónként, a fájlhoz kötöttek (azaz a fájl directory bejegyzésben jelentenek egy-egy flaget).

A NetWare rendszernek erényei mellett hiányosságai is vannak: Egyrészt nem tudja védeni a helyi adatokat (a munkaállomások Winchestereit), másrészt nem biztosít lehetőséget az adatok kódolására. Mint a legtöbb adatvédelmi rendszerben, itt is van egy vagy több kitüntetett jogú személy, ún. supervisor. Aki ilyen joggal rendelkezik, korlátlan hozzáférése van mindenhez, valamint tetszőlegesen jelenthet be más – akár supervisor jogú – személyeket is. Mivel a NetWare supervisor ugyanolyan felhasználó (neve és egyetlen jelszava van), ha valaki be akar törni egy Novell hálózatba, legokosabb, ha supervisor jogú nevet és jelszót keres.

Az utóbbi időben megjelentek valódi adatvédelmi rendszerek is. Ezek többsége ún. menü-rendszer, ahol az egyes felhasználókhoz menük rendelhetők, s csak az ott meghatározott parancsokat adhatják ki. Ilyen pl. a LE MENU ("Network and Single Use Hard Disk Manager") nevű rendszer. Komolyabb védelmet nyújt a SAFE-GUARD rendszer (nem tévesztendő össze a SOFTGUARD céggel), de igazán megbízható adatvédelmet jelenleg csak az EltSECRET rendszer ad.

LE MENU

A LE MENU — mint a fentebb leírt bővebb angol neve is mutatja — elsősorban fájlkezelő rendszer, kiegészítve menüvel. Több rétege van, ezeket különböző jelszavakkal lehet védeni. (A rendszer installálható úgy is, hogy egyáltalán ne kérjen jelszavakat.) Itt is van egy kitüntetett jelszó, a supervisoré, csak ezzel a jelszóval lehet kilépni a DOS-ba (a prompt-hoz). Külön jelszavak tartoznak a "DOS Services" és "Directory Management" funkciókhoz. Az első lényegében az alap DOS parancsokat engedi meg (FORMAT, BACKUP, RESTORE...) kényelmesebben, míg a második maga a fájlkezelő rendszer.

A menürendszer igen egyszerűen állítható össze (supervisor jelszó kell) és egyszerűen is kezelhető. Több mélységű menük alakíthatók ki (mindegyik szinten 26, ezeket a betűkkel lehet jelölni A-Z-ig). Egy menüpont lehet Program, Batch, Submenu ill. DOS services. A programnál meg kell adni a program pontos helyét (egység és elérési út), nevét és a pontos paramétereket is. A Batch indításoknál megenged speciális dolgokat is, melyek a DOS batch parancsaiban nem szerepelnek: ilyen pl. az a lehetőség, hogy egy program indítása előtt bekérhető a hozzá tartozó paraméter, belül át lehet váltani más menüre, stb... Ebből következik, hogy ezek a Batch-ek nem .BAT fájlba kerülnek, hanem a LE MENU maga dolgozza fel az eljárást lépésenként. A menük összeállításakor a supervisor bármelyik menü (almenü) esetén adhat jelszóvédelmet, ekkor az adott menü csak a jelszó ismeretében érhető el.

A felhasználók név szerint vannak nyilvántartva, azonban nem tartozik hozzájuk jelszó. (Csak egy indító menüt lehet rendelni az egyes felhasználókhoz). A rendszer indításakor (az AUTOEXEC.BAT-ban a LE MENU program paramétereként) adható automatikusan induló menü, ill. kezdő felhasználó. (pl. LEMENU User = 'LACI' MENU = 'L')

Fájlokat off-line kódolni a Directory Management menüből lehet. Meg kell adni egy jelszót (ez a fájlhoz tartozik), az oda-visszakódolás azonos (nincs külön Encrypt és Decrypt).

Komoly hibája a rendszernek, hogy nem indítható belőle tárrezidens (TSR) program (ui. a LE MENU minden program befejezésekor felszabadítja a teljes tárat). Emiatt ha valaki ilyen programot akar használni (akár a legegyszerűbb billentyűzet átdefiniálót, vagy pl. SideKick-et), annak hívását az AUTOEXEC.BAT-ba a LE MENU indítása elé kell tenni. (Ebből következik, hogy az minden felhasználóra azonos, nem lehet válogatni.)

A rendszer adatvédelmi problémái:

Nagyon kényelmetlen a felhasználók számára, hogy igen sok jelszót kell fejben tartani csupán a menükben való mozgáshoz is. Bár a jelszavak kódolva vannak, megkeresésük nem jelent komoly nyomkövetési problémát. A DOS-ba való kilépés csak supervisor jelszóval engedélyezett, ha valaki belép a DOS Services-be (márpedig igen sok dolgot csak onnan lehet megtenni, ezért sok felhasználónak ismernie kell az ehhez szükséges jelszót), ott van egy "Execute Command" menüpont, amivel tetszőleges program tetszőleges paraméterrel elindítható (ugyanaz, mintha a DOS promptnál lenne a felhasználó). A fájlok kódolása ugyanolyan nehézkes és problémás, mint a III. fejezetben említett siffrírozó programokkal, hiszen minden egyes fájlt manuálisan lehet csak kódolni, és minden fájlhoz ismerni kell az ahhoz tartozó jelszót. Legnagyobb problémája a hozzáférést ellenőrző rendszernek természetesen az, hogy saját rendszert indítva (floppy boot) nem indul el a LE MENU, tehát a felhasználó bármihez hozzáférhet.

A rendszer előnyei:

Könnyen kezelhető, többszintű menürendszert ad. Lokális hálózaton is használható. (Installálható az indítás floppyra is, hogy a Winchesterrel nem rendelkező hálózati állomásokról is lehessen a rendszertöltéskor a LE MENU-t használni.) A DOS Services menüben megtalálhatók a NetWare segédprogramjai (Login/Logout, Attach server, Spool, List users, Send msg) is. Használható ún. Projekt kód (erre ugyan semmilyen ellenőrzés nem történik, de statisztikákhoz használható). Bár a Directory Management kényelmes, szolgáltatásai elmaradnak más hasonló programok (pl. PcTools, Norton Commander) mögött.

SAFE-GUARD

A UTI-MACO cég SAFE-GUARD rendszere több változatban kapható: Az alaprendszer szolgáltatásai kibővíthetők a BACKUP opcióval (ez egy gyors mentő/visszatöltő program-pár), ill. kiegészíthető hardver kártyával, mely az adatvédelmet biztonságosabb szintre helyezi. Mivel az adatvédelem szempontjából a BACKUP érdektelen, ezzel az opcióval itt nem foglalkozunk.

A SAFE-GUARD alapvetően menürendszer, védelmi szolgáltatásai – néhány kivételtől eltekintve – csak a menün belül érhetők el. A rendszerben – mint a legtöbb adatvédelmi rendszerben – van egy kitüntetett supervisor, itt SYSTEM a neve, s ez a felhasználó sohasem törölhető a rendszerből. A rendszer installálásánál a SYSTEM jelszava "System", természetesen ez az installálás után megváltoztatható. (Probléma: ha elfelejtik ennek a felhasználónak a jelszavát, újra kell installálni a SAFE-GUARD rendszert, s teljesen újra kell kezdeni feltölteni a menürendszert. Ez ellen némi védelmet nyújt, ha a rendszer alapfájlját – neve SM-MEN.DAT, később még lesz szó róla – egy rögzített és feljegyzett supervisor jelszóval kimentik lemezre. Innen visszatöltve a rendszer feléleszthető.)

A rendszer karbantartása a supervisor feladata. Ő határozza meg az alap paramétereket (ld. alább), ő állítja össze a felhasználói menüket. Joga van kilépni a DOS-ba (a prompt-hoz) is.

A SAFE-GUARD rendszerben lehetőség van néhány nagyon fontos általános rendszerparamétert beállítani. Ezek közül adatvédelmi szempontból három lényeges: be lehet állítani a minimális jelszó-hosszt (1-8 karakter), le lehet tiltani a képernyő kiírását nyomtatóra (PrtScreen), valamint korlátozni lehet akár a Winchester, akár a floppyk használatát. Utóbbiaknál megadható, hogy írhatók-e ill. olvashatók-e. Ha egyszer a supervisor beállította ezt, akkor a tiltás mindenkire (rá is) vonatkozik mindaddig, míg meg nem változtatják ezt. (Nem lehet tehát felhasználók szerint szelektálni.) Sajnos az ezt figyelő SM-RES.COM program – a UTI-MACO szerint – sok más tárrezidens programmal összeveszhet. Ilyen esetre azt ajánlják, hogy az SM-RES-t ne töltsék be (azaz ilyen TSR programok mellett a floppy ill. Winchester védelem nem használható).

A UTI-MACO azt ajánlja – s ezt érdemes is megfogadni –, hogy a felhasználókat definiáló programot (SM-DEF) a supervisor mindig floppyról indítsa. Amikor a supervisor új felhasználót jelent be (max. 30 felhasználó lehet), nemcsak a menürendszert adhatja meg, de kijelölhet számára egy saját alkönyvtárat is. Az alkönyvtárban korlátozhatja a felhasználható lemezterületet. Ebbe az alkönyvtárba kerül a felhasználó bejelentkezés után.

(Problémás — nem az adatvédelem hibája —, hogy olyan overlayes programoknál, ahol az overlay nem tölthető PATH-ból, minden felhasználó alkönyvtárába be kell másolni azt. Ilyen pl. a régi Wordstar WS.OVL) Amennyiben a bejelentkezés időpontjában az alkönyvtárban levő fájlok összes helyfoglalása nagyobb a felhasználói limitnél, a rendszer automatikusan meghív egy segédprogramot, amelyikkel törölni lehet fájlokat onnan. Csak ha a törlések után a limitnél kisebb a helyfoglalás, akkor dolgozhat a felhasználó. Ez a fajta helyfoglalási korlátozás csak részleges védelmet ad, mivel a felhasználó más alkönyvtárban (pl. a saját könyvtárán belül létrehozott alkönyvtárban is) tetszőlegesen foglalhat helyet, ill. saját helyén is feltöltve a lemezt úgy hagyhatja a gépet, megakadályozva mások munkáját. (A SAFE-GUARD csak a supervisort engedi meg bejelentkezni, ha a Winchesteren 5Kb-ánál kevesebb szabad terület van csak.) A Winchester partíciók (már ahol van, az alap C: kivételével) lezárhatók, ill. bizonyos felhasználókhoz rendelhetők. A supervisor jogosult a felhasználók aktuális jelszavának megnézésére is.

A menürendszer összeállítása interaktív, ún. mask-editor segítségével történik. A rendszerben egyenként legfeljebb 96 menü, batch eljárás ill. segélyképernyő (HELP) definiálható, ezek együttes száma azonban nem lépheti túl a 150-et.. A teljes képernyőn akárhova helyezhető szöveg ill. kiválasztható menüpont, természetesen a színezés is változtatható. Egy menüben max. 30 menüpont adható meg, ezek bármelyike lehet: almenü, programindítás, batch, visszalépés (előző vagy főmenübe), kilépés a DOS-ba vagy kijelentkezés (logoff). A menük 7 szinten ágyazhatók egymásba. Programnál (és batch-nél) nemcsak a paraméterek definiálhatók (ezek között lehet ?-lel jelzett, amelyik helyen indítás előtt aktuális tartalom adható meg), de bármelyik programnál szabályozható, hogy abból ki szabad-e lépni a DOS-ba vagy sem. (Természetesen ennek csak olyan programoknál van jelentősége, amelyek megengedik a DOS SHELL használatát.) A DOS letiltását érdekesen oldja meg: Ilyenkor a SAFE-GUARD a program indítása előtt az Environment-ben a COMSPEC-et egy saját üres programra irányítja: így ha egy program DOS SHELL-t hív, üres futás lesz belőle. (Nem lesz sikeres a letiltás olyan "buta" programnál, amelyik nincs figyelemmel a COMSPEC aktuális értékére, hanem automatikusan a COMMAND.COM-ot indítja. Ilyen esetben semmi akadálya nincs a DOS-ba való kilépésnek.)

A belül megadott batch eljárások speciálisak, ezeket a SAFE-GUARD saját rendszerfájlijában őrzi, s csak futtatáskor keletkezik belőle — a felhasználó alkönyvtárában — egy \$\$\$\$\$\$.BAT nevű fájl. Ennek előnye, hogy ezeket az eljárásokat a felhasználók nem módosíthatják. Hátránya viszont, hogy max. 22 sort tartalmazhatnak.

Fontos opciója a SAFE-GUARD-nak az a lehetőség, hogy az egyes felhasználókhoz a főmenün kívül definiálható egy kezdő és egy végeljárás. Ezek lehetnek programok vagy batch feldolgozások. A kezdő eljárás bejelentkezés után azonnal elindul, s csak annak befejeződése után kerül a felhasználó a tényleges menürendszerbe. A végeljárás a kijelentkezéskor fut le. A SAFE-GUARD védelmi képességei annyira behatároltak (ld. alább), hogy lényegében minden felhasználónak munka előtt fel kell oldani a védettségeket (alkönyvtár, fájl rejtés, szükség szerint off-line kódolás), munkája befejeztével pedig ismét védetté kell tenni ezeket. Mivel a védelmeket egy-egy segédprogram állítja, megfelelő batch fájlokat összeállítva elő- és utóprogramnak, a felhasználó bejelentkezésekor ezek lefutnak, így a védelem megléte a felhasználó számára nem jelent plusz tevékenységet. A SAFE-GUARD biztonságossá is teszi ezt a módszert: ha ui. egy felhasználó nem jelentkezik ki, és a számí-

tógépet kikapcsolja, vagy új rendszert tölt, a rendszerindításkor az utoljára bejelentkezett felhasználó befejező eljárása automatikusan lefut, így biztosítva a megfelelő védettségi állapotot.

A SAFE-GUARD az adatfájlok védelmét alapvetően a DOS-ra bízta. Ez azt jelenti, hogy – az SM-H segédprogrammal – bármely fájl rejtetté (hidden), rendszerfájllá (system) ill. csak olvashatóvá (R/O) tehető. (Természetesen ezek vissza is állíthatók.) Hidden ill. System fájlok a normál DIR listákon nem jelennek meg, így – elvileg – a felhasználók nem is tudnak ezekről. Semmiféle védelem nincs azonban arra, hogy ezt a programot ki használhatja, így aki pl. kiléphet a DOS-hoz, az tetszőlegesen változtathatja bármely fájl attribútumait. Ugyancsak nincs semmi védelem azok ellen a programok ellen, amelyek figyelmen kívül hagyják, ill. maguk is változtatni tudják ezeket a jellemzőket. A SAFE-GUARD a fájlok szintjén csak a legegyszerűbb védelmet adja. (A kódolásról ld. később!)

Az alkönyvtárak védelme sokkal trükkösebb, de korlátozottabb is. A SAFE-GUARD úgy tud elrejtetni alkönyvtárat, hogy azt még a PcTools vagy Norton Utility sem látja. Az ilyen alkönyvtár neve ##-kal kezdődik, s a rendszer titkosításkor ennek első # karakterét pont-ra (.), a másodikat bináris nullára cseréli közvetlenül a directory bejegyzésben. Visszaállításkor a bejegyzésbe visszaírja a két # jelet. A védelemnek van azonban gyengéje is: egyrészt van olyan program (az Explorer), amelyik megmutatja ezt a speciális bejegyzést és vissza is tudja állítani a helyes állapotra, másrészt ha valaki tudja, milyen nevű alkönyvtárat kell visszahozni (a név a két karakter után 1-6 hosszú lehet) és elindíthatja az SM-H programot a megfelelő paraméterrel, az bármilyen ilyen alkönyvtárat vissza tud állítani. (Ha a felhasználó manuálisan végzi az alkönyvtár elrejtését, és utána elfelejti a nevét, csak az Explorerben reménykedhet, hogy megtalálhatja könyvtárát.)

A SAFE-GUARD kétféle adatkódolást tesz lehetővé: off-line fájl kódolást az SM-C segédprogrammal, ill. on-line Winchester-kódolást a SAFE-GUARD PLUS hardverkiegészítő kártyával. Az off-line kódolás menete hasonló a III. fejezetben leírtakhoz, azaz a megfelelő fájlokat munka előtt dekódolni kell (decrypt), munka után ismét visszakódolni (encrypt). A SAFE-GUARD azonban ezt a felhasználó számára igen kényelmesen végzi: nem kell külön-külön fájl jelszavakat megjegyezni, mivel a kódolás a felhasználó saját jelszavával történik. Az SM-C-vel egy alkönyvtárban egyszerre több fájl is kódolható (akár *.* paraméter is adható). Ha a dekódolást a felhasználó előtét eljárásába, a kódolást a logoffkor lefutó utóeljárásba teszik, viszonylag kényelmesen dolgozhat: a kódolások számára transzparenssé történnek, nem kell beavatkozni semmibe. Természetesen ennek az eljárásnak megvannak a maga hátulütői is: egyrészt maga a kódolás mindig az összes fájlra lefut, függetlenül attól, hogy éppen melyikkel akar a felhasználó dolgozni, ráadásul elég időigényes is: Turbo XT-n 1 Mbájtnál több kódolása több, mint másfél percre tart. (Ha azonban összehasonlítjuk ezt a SUPERKEY DES algoritmusának 22 és fél perces "gyorsaságával", látható, hogy az algoritmus még mindig elég egyszerű. Hasonlít az algoritmus annyiban, hogy itt is 8 bájtontként kódol a program.) A kódolás mindig biztonságosan, fájl másolással történik. (Ennek hátránya azonban, hogy kizárólag akkor hajtható végre, ha a fájl méretének megfelelő szabad terület van a lemezen, így igen nagy fájlok esetén – az idő mellett – nem mindig használható megoldás.) A felhasználó jelszót csak akkor cserélhet, ha előtte minden adatfájlját dekódolta, hiszen különben nem tudná azokat helyesen visszakódolni. A rendszer legnagyobb hátránya azonban a jelszófüggősége: egy bizonyos adatfájlt csak

akkor lehet több felhasználó között megosztani (pontosabban megengedni, hogy más-más időpontban használni tudják), ha ugyanaz a jelszavuk. Ez az off-line kódolást teljesen személyes használatra korlátozza.

Az on-line kódoláshoz a hardverkiegészítő is kell. Mielőtt ezt installálnák, egy speciális segédprogrammal a teljes Winchestert át kell kódolni. (A kódolásban szerepel a hardverkártya sorozatszáma, a SER# is, tehát az egyes kártyák különbözőképpen kódolnak.) Miután a hardvert installálták, az az összes Winchester I/O műveletnél dolgozik: kiírásnál kódol, olvasásnál dekódol. A kódolás fizikai adatszektoronként történik. Az on-line kódolás előnye gyorsasága, valamint az, hogy ha a Winchestert ellopják, akkor sem olvasható a rajta levő adat. A megoldásnak azonban komoly hiányosságai is vannak: egyrészt lopás ellen igen korlátozott a védelem (hiszen a kódoló kártyát is vele lophatják), másrészt a Winchestert egységesen kódolja, azaz nem lehet felhasználónként szeparálni a kódolást. Ez azt jelenti, hogy egy ilyen kártyával ellátott gépen ha egy felhasználó jogosult dolgozni, akkor ő a teljes Winchestert mindig dekódolva látja. (A UTI-MACO megjegyzi, lehet, hogy bizonyos programvédelmekkel a hardver nem működik együtt. Ha ilyet tapasztal a felhasználó, nem használhatja az on-line kódolást.)

A hardverkártya még egy lehetőséget ad: le lehet vele tiltani a floppyról történő rendszerbetöltést. Ezzel el lehet érni, hogy minden rendszertöltés (boot) a Winchesterről történjen, s ha ott az AUTOEXEC.BAT (ill. a SAFE-GUARD esetében a SAFE-GRD.BAT) tartalmazza az adatvédelmi rendszer indítását, a felhasználó bekerül a menürendszerbe, kötelezően logonolnia kell.

Az automatikus rendszerindítást a SAFE-GUARD igen ötletesen oldja meg: Ismert, hogy a DOS rendszerindításkor automatikusan elindítja az AUTOEXEC.BAT-ot (amennyiben az létezik a főkönyvtárban). Ennek az eljárásnak a kockázata, hogy az AUTOEXEC.BAT futása Ctrl-Break segítségével "lelőhető". A SAFE-GUARD ezért indítását nem oda teszi, hanem egy SAFE-GRD.BAT fájlba, s installációkor átírja a CONFIG.SYS-t úgy, hogy az induljon el. Ehhez a következő sort kell a CONFIG.SYS-be tenni:

```
SHELL = C:\COMMAND.COM /C SAFE-GRD.BAT
```

Látszólag nincs sok különbség, hogy a COMMAND.COM mit indít, az AUTOEXEC-et vagy mást, azonban egy nagyon fontos eltérés, hogy ha most a rendszerindításkor valaki Ctrl-Break-kel leállítja a SAFE-GRD.BAT futását, a parancsprocesszor befejezi működését. Mivel azonban alatta már nincs másik parancsprocesszor, terminálásakor az egész rendszer kiakad, a számítógép "lefagy", csak rendszertöltés indítja újra.

A SAFE-GUARD adatvédelmi rendszer elég sok adatot tárolhat a felhasználó munkájáról. Nemcsak a be- és kijelentkezéseket, valamint az érvénytelen kísérleteket gyűjti, de bármely programnál megadható, hogy indításáról bekerüljön-e bejegyzés az ún. Log-data-ba vagy sem. Emellett az SM-UTI lemezkezelő segédprogramot lehet úgy indítani, hogy minden műveletet (másolás, törlés, stb...) feljegyezzen.

A SAFE-GUARD lemezkezelő segédprogramja, az SM-UTI elég kényelmesen használható. Mivel azonban rengeteg hasonló program kapható, nem érdemelne figyelmet, ha adatvédelmi szempontból nem rendelkezne egy igen fontos opcióval: a program indításakor limitálható használata. Az SM-UTI használatakor letiltható a Winchesterhasználat (azaz a

felhasználó csak saját floppyjain dolgozhat, már amennyiben a floppy nincs teljesen letiltva), letilthatók fájlcsoporthok (kiterjesztés megadásával), letiltható a törlés, átnevezés, másolás, nyomtatás. Mindezek akkor érdekesek, ha a menürendszerben van az SM-UTI indítás, így ui. a felhasználó lehetőségei behatárolhatók. (Éppen emiatt ajánlja a fejlesztő, hogy csak az ő segédprogramját használják.)

A rendszer – amennyiben a SAFE-GUARD PLUS hardvert is használják – megbízható védelmet nyújt, ha a felhasználókat teljesen szigorúan korlátozzák. Ennek alapvető feltétele, hogy semmilyen módon nem szabad kiengedni a DOS-ba, kizárólag a menürendszeren belül dolgozhat. A rendszer alapfilozófiája miatt ha a legkisebb rés van a rendszeren, lényegében teljesen átjárhatóvá válik. (Ha egy felhasználó kikerül a DOS-ba, vagyis szabadon indíthat programokat, többé semmi nem korlátozza a programok – köztük a SAFE-GUARD segédprogramjainak – használatában. Tulajdonképpen csak az korlát, ha nem tudja egy rejtett alkönyvtár nevét – s esetleg nem tudja megkeresni Explorer-rel –, ill. az off-line kódolt fájlokat nem tudja dekódolni, ha más a jelszava.)

A rendszer adatvédelmi problémái:

Mint fentebb említettük, amennyiben egy felhasználó kijut a menürendszerből, semmi nem korlátozza tovább. Ebből az is következik, hogy hardverkiegészítés nélkül floppyról saját rendszert indítva csak a rejtett alkönyvtárak jelentenek problémát, a többi (hidden, system fájlok) a legegyszerűbb segédprogramokkal kikerülhetők. A SAFE-GUARD az ún. environmentet nem kezeli, ezért nem lehet PATH-t változtatni. Ha használni akarnak PATH-t, azt az indító SAFE-GRD.BAT-ba kell tenni. Ugyanemiatt az egyes menühívások között nem őrződnek meg az esetleg használt SET változók, így azokat a programokat, melyek ilyeneket használnak (pl. fordítók), batch eljárásba kell ágyazni.

Mivel a rendszer főkönyvtárat nem véd, valamint fájlvédelme sincs megfelelő, még hardverkiegészítő mellett sem tudja abszolút biztosítani az adatvédelmi rendszer indítását: ha egy felhasználó editorral elérheti akár a CONFIG.SYS-t, akár a SAFE-GRD.BAT-ot (s a legtöbb editorból az indított fájl editálásán kívül más is módosítható), ezeket megváltoztatva kikapcsolhatja a rendszer indítását.

A legkomolyabb probléma azonban – mint szinte minden adatvédelemnél – a jelszókkal van. A SAFE-GUARD minden adatát (alap rendszerparaméterek, felhasználói adatok, menük) az SM-MEN.DAT fájlban tárolja. Az első, ami feltűnik, ha belenézünk a fájlba, hogy a fájl legelején karakteres formában (!!!) ott találjuk a legutoljára bejelentkezett személy jelszavát. (Ez normál használatban nem zavaró, hiszen a felhasználó saját jelszavát amúgy is tudja, azonban hardverkiegészítő nélkül floppyról történő rendszerindítással máris szerezhető egy jelszó.) A nevek és jelszavak ugyan kódolva vannak, de nagyon egyszerű módon. Az alábbiakban láthatjuk azt a rutint, mely mindkét adat dekódolását végzi:

```

.....:4EF1  55      PUSH  BP
.....:4EF2  8BEC    MOV   BP,SP
.....:4EF4  55      PUSH  BP
.....:4EF5  E90000  JMP   4EF8
.....:4EF8  4C      DEC   SP
.....:4EF9  4C      DEC   SP
.....:4EFA  B80100  MOV   AX,1
.....:4EFD  50      PUSH  AX

```

```

.....:4EFE B80800 MOV    AX,8
.....:4F01 59      POP    CX
.....:4F02 91      XCHG   CX,AX
.....:4F03 2BC8    SUB    CX,AX
.....:4F05 7D03    JGE     4F0A
.....:4F07 E97B00 JMP     4F85
.....:4F0A 41      INC     CX
.....:4F0B 8946FC MOV    [BP-04],AX
.....:4F0E 51      PUSH   CX
.....:4F0F 8A4608 MOV    AL,[BP+08]
.....:4F12 32E4    XOR     AH,AH
.....:4F14 0BC0    OR      AX,AX
.....:4F16 7503    JNZ     4F1B
.....:4F18 E93100 JMP     4F4C
.....:
.....:
.....:4F4C C47E04 LES     DI,[BP+04]
.....:4F4F 8B46FC MOV    AX,[BP-04]
.....:4F52 03F8    ADD     DI,AX
.....:4F54 06      PUSH   ES
.....:4F55 57      PUSH   DI
.....:4F56 C47E04 LES     DI,[BP+04]
.....:4F59 8B46FC MOV    AX,[BP-04]
.....:4F5C 03F8    ADD     DI,AX
.....:4F5E 268A05 MOV    AL,ES:[DI]
.....:4F61 32E4    XOR     AH,AH
.....:4F63 050F00 ADD     AX,0F
.....:4F66 50      PUSH   AX
.....:4F67 8B46FC MOV    AX,[BP-04]
.....:4F6A 050300 ADD     AX,3
.....:4F6D B90900 MOV    CX,9
.....:4F70 F7E9    IMUL    CX
.....:4F72 59      POP     CX
.....:4F73 91      XCHG   CX,AX
.....:4F74 2BC1    SUB     AX,CX
.....:4F76 5F      POP     DI
.....:4F77 07      POP     ES
.....:4F78 268805 MOV    ES:[DI],AL
.....:4F7B 59      POP     CX
.....:4F7C 49      DEC     CX
.....:4F7D 7406    JZ       4F85
.....:4F7F FF46FC INC     W,[BP-04]
.....:4F82 E989FF JMP     4F0E
.....:4F85 E90000 JMP     4F88
.....:4F88 8BE5    MOV     SP,BP
.....:4F8A 5D      POP     BP
.....:4F8B C20600 RET     6

```

(A kipontozott helyen az ellentétes rutin, a kódolás található.)

A rutin — látszólagos bonyolultsága ellenére — nem tesz mást, mint a név vagy jelszó első bájtyából kivon (hexa) 15-öt, a másodikból 1E-t, a harmadikból 27-et, a negyedikből 30-at, az ötödikből 39-et, a hatodikból 42-t, a hetedikből 4B-t és a nyolcadik bájtból 54-et:

	e1	e2	e3	e4	e5	e6	e7	e8
Sub hexa	15	1E	27	30	39	42	4B	54
	d1	d2	d3	d4	d5	d6	d7	d8

Nézzük meg mindezt a supervisor adataival (a jelszó 6 bájtós, az utolsó két bájt tartalma érdektelen):

```
.....:1930 00000688979AA49E-AFBCB00668979AA4 .....h...
.....:1940 9EAF7C8C0668979A-A49EAF7C8C002020 ..|..h.....|..
```

ill. dekódolva:

```
.....:1930 0000067379737465-6D715C0653797374 ...systemq\.Syst
.....:1940 656d313806537973-74656d3138002020 eml8.Systeml8.
```

(A név, ill. jelszavak előtt található hosszuk, jelen esetben mindháromszor 6. A név utáni első jelszó az ún. előző jelszó, ezt azért tárolja a rendszer, mert jelszócserénél az aktuális, ill. előző jelszót nem lehet megadni.)

Mivel a SAFE-GUARD rendszerben a supervisor neve rögzített, nem változtatható, ezért az SM-MEN.DAT-ban is rögzítetten, mindig a fenti kódsorozattal szerepel. Ezután, ha valaki meg akarja tudni a supervisor jelszavát, nincs más teendője, mint megkeresni a fájlban a (hexa) 88979AA49EAF kódsorozatot, majd ennek kezdetétől 18 (hexa 12) bájtnyira kiírni az aktuális jelszó kódolt alakját (a hossz az előző bájton található). Ha ezután manuálisan elvégzi az előbb leírt dekódolást (kivonva a megfelelő értékeket), már meg is kapta a keresett jelszót.

A SAFE-GUARD másik komoly hibája, hogy kifejezetten egyéni, szeparált felhasználói munkára tervezték. (ld. az off-line kódolást) Lokális hálózaton nem használható, dokumentációja meg sem említi az ezzel kapcsolatos bonyodalmakat. (DOS alatt futó hálózati rendszereknél – pl. ORCHID PC NET – elvileg a serverbe betehető a PLUS hardver kódolója, azonban Novell NetWare esetén már nem használható.)

A rendszer előnyei:

Ha valahol néhány felhasználó teljesen szeparáltan dolgozik, a rendszerrel kényelmes menükörnyezet teremthető számukra. A rendszer az átlagfelhasználó munkáját pontosan szabályozza, még kódolt fájlok esetén is különösebb procedura nélkül dolgozhat: a többi Encrypt programmal ellentétben nem kell sok jelszót megjegyeznie, csak a sajátját. Mind a fájlok ill. alkönyvtárak elrejtése ill. elérhetővé tétele, mind a kódolás automatizálható az előtét és utóeljárással. Ha csak az SM-UTI használatát engedik meg, a felhasználók tevékenysége jól dokumentálható lesz. Fontos, hogy a rendszer hardverkiegészítővel lehetővé teszi a floppyról történő rendszerindítás megakadályozását. A felhasználó maga cserélheti jelszavát, sőt a supervisor beállíthatja, hogy mennyi időnként kötelező ezt megtenni.

ElitSECRET

Az ELTISOFT cég terméke kiemelkedik az adatvédelmek közül. Ez az első olyan rendszer, mely megpróbál zárt, igazi védelmet kialakítani akár egyedi gépeken, akár hálózatokkal együttműködve. Több szintje van, kezdve a legegyszerűbb hozzáférésvédelemtől a hardverrel támogatott ellenőrzésig. A szintek:

- hozzáférésvédelem
- információvédelem

- menürendszer
- hardverkiegészítéssel személyi azonosítás

Az egyes szintek egymásra épülnek, a legalsó szinttől eltekintve mindegyik opcionális. A rendszer teljes kiépítésben (megfelelő fájlkódolásokkal, menürendszerrel, hardverkiegészítővel) teljes adatbiztonságot képes nyújtani. A rendszer alapfilozófiája a maximális biztonságra törekvés. Ebben a rendszerben valósul meg legelőször, hogy a felhasználók jelszavai nincsenek letárolva, így még a legjobb nyomkövetéssel sem lehet azokat megkapni. Mivel a rendszer szolgáltatásai a védelmi szintekkel vannak összefüggésben, azok szerint fogjuk a rendszert áttekinteni.

Hozzáférésvédelem:

A rendszer hierarchikus hozzáférést biztosít az adatokhoz. Ez úgy valósul meg, hogy a védeni kívánt fájlokat ill. alkönyvtárakat csoportokba (ún. szintekre) kell sorolni. 64 fájl szint és 127 alkönyvtár szint jelölhető ki. Ezután minden felhasználónál meg kell adni, mely szintekre jogosult hozzáférni (olvasásra ill. írásra). Mivel bármely felhasználónál a szintek bármely kombinációja megengedett, a lehetséges kombinációk száma minden igényt ki tud elégíteni. Az írásvédelem mindenféle módosítás (írás, törlés, átnevezés, létrehozás, attribútum, dátum átírás) ellen véd. Az EltSECRET védelme egy tárrezidens programon alapul. Ez "ráül" a DOS funkcióhívásokra (INT 21), és minden fájl műveletet ellenőriz. Ennek a megoldásnak köszönheti a rendszer, hogy a hozzáférésvédelem teljeskörű, nem egy menürendszerhez kötött. (Így a felhasználók nyugodtan kiengedhetők a DOS prompthoz a menürendszerből is, az előlük elzárt fájlokhoz ill. alkönyvtárakhoz nem tudnak hozzáférni.) Mivel nemcsak adatfájlok, de programok is védetté tehetők, a menürendszeren kívül is letilthatók bizonyos programok. (A felhasználó számára tiltott programokat hiába próbálja meg elindítani, a DOS nem fogja ezeket megtalálni, "Access denied" hibaüzenet jelenik meg a képernyőn.)

A fájlok védelme mellett minden felhasználónál megadható, jogosult-e floppyt használni, vagy sem. A floppyhasználatot le lehet tiltani teljesen (ebben az esetben az EltSECRET egyoldalas, írásvédett, üres lemezt szimulál), vagy csak olvashatóvá lehet tenni. (Sajnos floppy-letiltásnál a kulcslemez védelmek nem tudnak működni – kivéve az EltGUARD védelmet. Utóbbi kivételezettsége talán érthető, mivel ugyanaz a fejlesztője, mint az EltSECRET-nek.)

A hozzáférésvédelem önmagában nem elegendő, hiszen ha csak ezt használnák, bárki saját floppyról saját rendszert indítva kikerülhetné a védelmi rendszert.

Információvédelem:

Az EltSECRET adatvédelmi rendszer legfinomabban kidolgozott része az adatok titkosítása. Felépítésén látszik, hogy tervezésekor minden részletre ügyelve teljesen zárt védelmet kívántak kialakítani. A rendszernek ebben a részében nem található kiskapu, nem lehet ügyeskedni, jelszavakat megtudni. Hogyan éri el ezt a tökéletes biztonságot a rendszer?

Az EltSECRET lehetőséget biztosít arra, hogy fájlokat, fájl csoportokat on-line kódoljanak. A kódolás transzparens a felhasználói programok számára, azaz nem kell a programokon módosítani ahhoz, hogy a kódolt adatfájlokkal dolgozni lehessen. Az eddig tárgyalt

adatkódolásoktól eltérően a felhasználónak nem kell – manuálisan vagy félautomatizáltan – munka előtt adatait dekódolnia, munka után kódolnia. A lemezen (Winchesteren vagy floppy-n) mindig kódoltan helyezkednek el a kijelölt adatfájlok, a kódolás (íraskor) ill. dekódolás (olvasáskor) teljesen automatikusan, a felhasználó számára észrevétlenül megy végbe. Ennek a megoldásnak egyik előnye, hogy mindig csak a szükséges adatrészt kell kódolni, másrészt hálózaton is alkalmazható: a serveren mindig kódolva vannak az adatok – minden felhasználó számára –, az átalakítás mindig a felhasználói munkaállomás memóriájában történik. (Ez egyrészt tehermentesíti a servert a kódolástól, másrészt a hálózat elemein – vezetékein – csak kódoltan futnak az információk.) Az on-line kódolás – a SAFE-GUARD totális hardvermegoldásától eltérően – nem egységes a teljes lemezre, hanem csak a kijelölt fájlokra vonatkozik. Természetesen a kódolásokhoz valamilyen jelsorozatot kell, ami biztosítja a visszafejthetetlenséget. Itt az EltSECRET teljes biztonságot ad: a kódoláshoz szükséges jelsorozat a felhasználó valódi jelszavából alakul ki. Itt válik igazán lényegessé az, hogy e rendszerben nincsenek letárolva a felhasználók jelszavai, csak azokról egy ellenőrző összeg. (Utóbbi a jelszavak véletlen elírása ellen véd.) Ha valaki nem tudja a teljes jelszót, hiába próbál belépni a rendszerbe (pl. nyomkövetéssel kiiktatva az ellenőrzést, vagy megkeresve az ellenőrző összeget, és olyan összegű jelszót beírva), az adatfájlokhoz hozzáférést kaphat, de az információt nem tudja kiolvasni, hiszen a rendszer az általa beírt – hibás – jelszó alapján fog dekódolni, azaz hibás adatot fog kapni. Az EltSECRET kikerüli a SAFE-GUARD azon buktatóját, hogy a felhasználói jelszóval való (ott off-line) kódolás miatt egy fájlhoz csak egy felhasználó férhet hozzá: a jelsorozat kialakításához a felhasználói jelszón kívül egy másik jelsorozatot is használ (ezek felhasználónként mások és mások, a rendszer tárolja őket), így más-más felhasználó más-más jelszóval is ugyanazt a jelsorozatot kapja a kódoláshoz, így az adatfájlokat helyesen tudják feldolgozni (hálózati környezetben akár egyidejűleg is).

Nemcsak adatfájlok, de programok is kódolhatók. Ez akkor hasznos, ha nem elégszenek meg az alap hozzáférési védelemmel, hanem biztosítani akarják, hogy illegális behatolási kísérlet esetén se lehessen bizonyos programokat elindítani. Mivel a valódi programkód csak a helyes dekódolással alakul ki, kódolt program az EltSECRET rendszeren kívüli elindítása – ha a DOS eleve nem utasítja vissza a kísérletet – majdnem biztos kiakadást okoz. (Bizonyos érzékeny programokat, pl. PcTools, Norton Utility célszerű ilyen módon elérhetetlenné tenni a felhasználók számára. Természetesen azok, akik jogosultak ezeket elindítani, gond nélkül használhatják őket.)

Az EltSECRET rendszer háromféle kódolási algoritmust tartalmaz, szintenként más és más használható a titkosításra. Ezek az algoritmusok bonyolultságban és időigényben különböznek. Az első a legegyszerűbb, ott 256 bájtos ciklussal XOR-ol, de az alaptábla az adott szint minden fájljára azonos. (Ez gyorsít, ugyanakkor csak olyan fájlokra javasolt, ahol nem cél a visszafejthetetlenség.) A második algoritmus bonyolultabb, fájlanként különböző, időigénye azonban még mindig elenyésző. E két kódolást a felhasználó munkája során nem veszi észre. A harmadik eljárás egy nagyon bonyolult, bitkeverést is tartalmazó módszer, időigénye ennek megfelelően nagy. Csak kisebb fájlok kódolására javasolt, különben nagyon lelassíthatja a munkát. (Időadatok: Turbo XT gépen 1 Mbájt átkódolása az első algoritmussal 55 sec, a másodikkal 1 perc 2 sec, a harmadikkal 9 perc 40 sec. Összehasonlításképp: 1 Mbájt normál másolása kb. 50 sec volt.)

Menürendszer:

Részben a felhasználói munka segítésére, részben a szabályozottab programhasználat céljára az EltSECRET is tartalmaz menürendszert. A menük felhasználónként definiálhatók, de csoportképzésre is lehetőség van. (Egy felhasználónál max. 10 menücsoport adható a saját parancsain kívül.) A menü lineáris, egymásba ágyazásra nincs lehetőség. Az egyes menüpontok magyarul írhatók meg, a képernyő alsó sorában látszik a parancs formája. (Segélyképernyő, HELP definiálható minden sorhoz.) Természetesen az egyes parancsok paraméterezhetők is, itt az EltSECRET többet ad az előzőekben tárgyalt menürendszerknél: a paraméternél limitálható a beírás. Lehet olyan paraméter, ahol csak egy karakter írható, lehet olyan, ahol szeparáló karakterek (szóköz, /-jel, stb.) nem írhatók, s lehet olyan paraméter, ahol bármi írható. (Természetesen a DOS Pipe parancskapcsolás tilos.)

A felhasználókhoz — a SAFE-GUARD-hoz hasonlóan — rendelhető bejelentkezéskor automatikusan induló program (vagy parancs). Ha egy felhasználónak nem adnak tényleges menüt, a program befejezésekor logoffolódik is. (Ezzel elérhető pl. hogy egy munkágyőn dolgozó, aki csak egy adott programot használhat, bejelentkezésekor azonnal abba a programba kerüljön, majd munkája végeztével automatikusan logoffolódjon is.)

Minden felhasználónál megadható, kiléphet-e a DOS-ba vagy sem. Ha igen, a menürendszerben is van ilyen főmenüpont, ha nem, semmilyen programból nem tud SHELL-t hívni. (A letiltást az alap rezidens program végzi.)

A menürendszerben a felhasználó az utolsó 10 parancsát visszahívhatja, ezzel segíti a menü pl. a hibásan elküldött parancsok gyors javítását.

Hardvervédelem:

Az EltSECRET rendszerben a hardverkiegészítés nem a kódolást szolgálja, hanem a biztonságot növeli. Egyrészt letiltható vele a floppyról történő rendszerbetöltés, másrészt lehetővé teszi, hogy a felhasználók ún. személyi azonosítót használjanak. Ez egy olyan kis eszköz, melybe a rendszer a bejelentkezéshez szükséges adatokat írja. Ezután a felhasználó csak úgy tud bejelentkezni, ha azonosítóját behelyezi az olvasóba, és úgy logonol. Ezzel nagymértékben nő az illegális belépések kockázata, hiszen nem elegendő valakinek a jelszavát megtudni (sajnos minden védelmi rendszerben a leggyengébb láncszem az ember), de a fizikai eszközzel is rendelkezni kell.

Az EltSECRET érdekes módon oldja meg a rendszer kötelező indulását: Mivel az indítás — hagyományosan — az AUTOEXEC.BAT-ban van, meg kell akadályozni ennek Ctrl-Break-kel való leállítását. Ezt a hardver úgy oldja meg, hogy a boot idejére fizikailag letiltja a billentyűzetet (port szinten). A tiltást csak a rezidens program (ill. menürendszernek csak a menüprogram) oldja fel. Ebből következik, hogy ha óvatlanul kitörlik az indítást, a számítógép rendszerindítás után nem fogad el a billentyűzetről semmit. Ilyenkor ki kell venni a hardverkátyát, s csak így indítva lehet a rendszert normális állapotba visszaállítani. Természetesen az EltSECRET-ben mind az AUTOEXEC.BAT-ot, mind a rendszer programjait védeni kell a felhasználók elől, célszerű ezeket egy speciális szintre helyezni, amelyik szintre írási jogot csak néhány személy kaphat.

Az EltSECRET rendszerben — hasonlóan, de nem ugyanolyan módon, mint más védelmeknél — ki kell jelölni egy személyt, az ún. supervisort (adatadminisztrátort), aki a rend-

szert felügyeli. Ennek a személynek azonban nincs speciális jogosultsága a rendszerben (nem tehet meg mindent), csak annyi a feladata, hogy ő állítja össze a rendszert. Az EltSECRET-ben a védendő fájlok, alkönyvtárak kijelölése, a felhasználók megadása, a menürendszer leírása nem speciális programmal történik, hanem a supervisor tetszőleges editorral vagy szövegszerkesztővel ún. leíró fájlokat (közönséges ASCII szövegfájlokat) készít. Mivel pl. a felhasználókat leíró fájl tartalmazza a nevek mellett a jelszavakat is, ezeket a fájlokat szigorúan floppyn kell tárolnia, s csak onnan szabad használni. (Ez a megoldás biztonságos, hiszen a lemez elzárható biztos helyre, másrészt bármikor utána lehet nézni elfelejtett jelszavaknak.) A leíró fájlokból speciális programok készítik el a rendszerfájlokat (egy az alaprendszerét, egy a menürendszerét). Az EltSECRET a supervisor jogát azzal befolyásolja, hogy a generáló programok indítási jogát csak neki adja meg. Ez nem jelszóval történik, hanem úgy, hogy a generáló programokat csak floppyról szabad indítani, s azok védettek, tehát indításukhoz kell az EltSECRET rendszer kulcslemeze. (Mondanunk sem kell, hogy a védelem az EltGUARD másolásvédelmi rendszer.) Természetesen minden rendszer egyedi, ("bedrótózva" a sorozatszám és a vevő neve), így az egyik rendszerbe nem lehet máshonnan származó generáló programokkal belenyúlni.

A supervisor a felhasználóknál nemcsak jogokat adhat meg, a floppyhasználatot szabályozhatja, de lehetőség van a bejelentkezéseket időben limitálni (lejárat, expiration date). A rendszer érdekesen oldja meg az ellenőrzést: bejelentkezéskor az aktuális rendszerdátum nem lehet korábbi, mint a rendszerfájl generálásának dátuma. Ha a supervisor időről időre újragenerálja ezeket – függetlenül attól, hogy módosultak-e vagy sem –, a lejárt jogú felhasználó nem léphet be (mivel nem tudja a dátumot visszaállítani).

A felhasználók jelszavát a supervisor adja meg, ezt csak személyi azonosítóval rendelkező személy képes változtatni. (A rendszer saját fájljait abszolút módon védi.) A jelszó minimális hossza 10 karakter (max. 20 lehet), így inkább mondatokat, kifejezéseket érdemes használni, mint jelszavakat. (Külföldön a legújabb nagygépes rendszerek minimális jelszóhosszként 8-at ajánlanak. A szakirodalom az ilyen jelszavakat "password" helyett "passphrase"-nek írja.) Személyi azonosító esetén a felhasználó – akár a menürendszerből, akár külön segédprogrammal – változtathatja a jelszavát, azonban ennek különböznie kell a legutolsó hat jelszavától. Természetesen a jelszócsere semmilyen hatással nincs a kódolásokra, mivel az új jelszónak megfelelő félinformáció kerül bele a személyi azonosítóba.

Az EltSECRET rendszer hiányosságai:

Mivel hardverkiegészítés nélkül nem lehet megakadályozni, hogy egy felhasználó floppyról saját rendszert indítson, harver nélkül az EltSECRET sem tud teljes hozzáférésvédelmet biztosítani. Bár mind a programok indítása, mind az adatok olvasása (pl. DBF fájlok feldolgozása) kódolással megakadályozható, nem tudja biztosítani ilyen esetben a szándékos rongálást (pl. fájlok törlését). Az adatvédelmi rendszeren belül bizonyos programokkal (pl. Norton Utility, PcTools), melyek nem fájl I/O-val, hanem közvetlen lemezolvasással dolgoznak, bele lehet olvasni a fájlokba, akár módosíthatók is. (Természetesen kódolt fájloknál ezek a programok csak a kódolt tartalmat látják, az információvédelem biztosított.) Célszerű ezért ezeknek a programoknak a használatát erősen korlátozni. (Megj: A Norton Utility nem tud belépni letiltott alkönyvtárba, mivel a DOS Change Directory műveletével dolgozik. Ugyancsak nem tudnak ezek a programok fájlokat törölni, átnevezni, stb.) A leg-

több lemezkezelő program (XTREE, Norton Commander, 1dir, PathMinder, melyek fájlkezelést végeznek) szabályosan működik, azaz a védett fájlokhoz nem férnek hozzá. Az EltSECRET nem strukturálja a módosítási jogokat, aki írásra jogosult, az mindenféle módosításra, törlésre is. (A Novell NetWare-ben többféle ilyen jogkör különböztethető meg.) Jelszócserére csak személyi azonosító mellett van lehetőség, egyébként a supervisortól kell ezt kérni. A menürendszer nem strukturált, nincs lehetőség több mélységben kialakítani a felhasználói menüket. Nincs a rendszernek saját lemezkezelő segédprogramja, ezért a felhasználók valamelyik ismert programot választhatják. (Norton Comander, XTREE, PathMinder, 1dir)

A kódolásnak van egy hátránya: mivel a kódolási algoritmusban a fájlnev is szerepel, nem lehet minden további nélkül átnevezni fájlokat. (Helyette másolni kell.) Ez néhány programnál a védett fájlok kezelésekor problémát okoz, ezeket speciális előtétprogramokkal (FoxBASE esetén STARTFOX, a többinél STARTREN) kell indítani. Mivel ilyen esetben az átnevezések helyett másolás történik, lassítja a program futását. (A FoxBASE esetén egészen egyedi megoldással nincs másolás, ezért lassulás sincs.) Mivel nem minden programról ismert, hogyan menti a fájlokat, az EltSECRET installálása után célszerű megnézni, mely programokat kell így indítani. (A FoxBASE-en kívül biztosan ilyen a Word, Wordstar, AutoCAD.)

A rendszer — normál használat mellett — nem teszi lehetővé, hogy mentéseket (backup) kódoltan készítsenek a felhasználók. Mind a speciális backup programoknál (FastBack, PcBackup), mind a streamerre mentésnél olvasáskor dekódolás történik, a mentésre íráskor viszont — mivel ott nem fájl szintű a kiírás — nincs ismételt kódolás: így a mentéshordóóra mindig kódolatlanul kerül az adat. Létezik ugyan egy OFFCRYPT nevű előtétprogram ennek a problémának a kezelésére, de az nem része a kereskedelmi EltSECRET verziónak.

A rendszer előnyei:

A legfontosabb előnye minden más rendszerrel szemben, hogy valódi információvédelmet ad. A kódolás megoldása biztosítja, hogy az adatok fájlok tartalmát valóban csak a jogosultak ismerhessék meg. Mivel a működő rendszerben a jelszavak még kódolva sincsenek tárolva (csak a supervisor elzárt floppyján található meg), nyomkövetéssel vagy egyéb trükkökkel sem lehet információt lopni az EltSECRET-tel védett fájlokból. Lokális hálózatok esetében biztosítja a kódolt fájlokhoz való egyidejű hozzáférést, a kódolás mindig a munkaállomáson történik. (Nincs lehetőség adatleszívásra a hálózatról.)

A jelszavak — az igen nagy minimális hossz miatt — kevéssé kitalálhatók. A rendszer a szisztematikus próbálkozást érdekes módon gátolja: Hibás bejelentkezéskor a bejelentkezés után kis ideig várni kell, míg a következő kísérlet kezdhető. Ez első hibánál kb. 2 másodperc, a második után 4, a harmadik után 8, stb. Mivel ez exponenciálisan növekszik, igen hamar kivárhatatlan lesz a következő próbálkozásig kivárandó idő. A jelszavak valódi cseréjét biztosítja (személyi azonosítónál), hogy a megelőző hat jelszó nem használható.

Bár a negatív oldalon említettük a kódolás fájlnevekhez kötését, van a megoldásnak egy igen trükkös haszna is: ha valaki saját kódolt adatfájlját átnevezi, az abban levő információ csak akkor lesz hozzáférhető, ha előtte visszanevezik. Amennyiben az eredeti név nem ismert, még a supervisor sem tudja a tartalmat kiolvasni a fájlból. (Egyéni titkosítás.) A

módszer egyetlen hátulütője: ha a felhasználó elfelejti az eredeti nevet, többé számára sem lesz hozzáférhető az információ.

Az EltSECRET speciális lehetősége az ún. time-out. A legtöbb adatvédelmi rendszer nem nyújt védelmet a feledékeny felhasználók ellen. Ha ui. valaki bejelentkezik, dolgozik, majd úgy hagyja ott a gépet, hogy elfelejt kijelentkezni, bárki leülve a helyébe elérheti mindazt, amire az adott felhasználó jogosult volt. Hogy ezt részben védje, az EltSECRET-ben megadható egy időlimit (3 és 30 perc között), mely után — ha nem tapasztalt aktivitást a gépen — az adatvédelmi rendszer automatikusan logoffolja a felhasználót. Az időlimit helyes megválasztása növelheti a védelem biztonságát.

A felhasználó biztonságérzetét növeli, hogy bejelentkezéskor a rendszer kiírja neki a legutolsó logon időpontját. Ennek az a jelentősége, hogy így a felhasználó azonnal látja, ha valaki a nevében dolgozott a számítógépen. (Ez a megoldás 1-2 éve ismert a nagygépes védelmi rendszerekben. Érdekes, hogy pl. a Novell nem vette még át ezt a módszert.)

A rendszerben a supervisor jogok nem egy adott személyhez és jelszóhoz vannak kötve, hanem programhoz és fizikailag a kulcslemezhez. Ennek elzárásával megoldható, hogy a rendszerben ne legyen olyan kitüntetett személy, aki korlátlan hozzáférési és módosítási joggal rendelkezik. Megfelelő szervezéssel megoldható az is, hogy a supervisor jogait is korlátozzák. Bár elvileg a leíró fájlból ismerheti az összes jelszót, személyi azonosítók használata esetén ez számára sem elég a más nevében történő bejelentkezéshez, ekkor ugyanúgy korlátozott személy lehet, mint bármely más felhasználó.

A felhasználói bejelentkezések — szintén csak a supervisor számára elérhető, védett programmal — listázhatók, statisztikák készíthetők. (Ebben az eltöltött idő is szerepel.) Természetesen az érvénytelen logon kísérleteket is gyűjti a rendszer, ezek is listázhatók.

A rendszer előnye, hogy tökéletesen együttműködik a lokális hálózatokkal. Mivel az EltSECRET alapfilozófiája szerint nem tárol jelszót, nem hajlandó a hálózatokkal azonos jelszavakkal dolgozni. Ez azt jelenti, hogy a hálózati rendszer jelszavait (Novell NetWare esetén csak a felhasználó hálózati jelszavát) is tudni kell az adatvédelmi jelszó mellett. NetWare esetén mindig először kell bejelentkezni (login) a hálózatba, s utána (logon) az EltSECRET adatvédelmi rendszerbe.

Ez az egyetlen olyan PC-s adatvédelmi rendszer, ahol megfelelő kiépítettséggel (hardverrel és pontosan definiált menürendszerrel) valóban biztos és teljeskörű adat- és információvédelem alakítható ki. A rendszer legnagyobb előnye, hogy mindezt úgy teszi, hogy az a felhasználó — és a felhasználói programok — számára láthatatlan. (Nincs komolyabb korlátozás sem a programok, sem a közös adatállományok használatával kapcsolatban.)

Bár nem szorosan az adatvédelemhez kapcsolódik, de meg kell említenünk egy komoly előnyét a rendszernek: Magyarországon ún. site licenc-szel forgalmazzák, ami azt jelenti, hogy egy telephelyen elegendő egyetlen példányban megvásárolni, azt ott tetszőleges számú gépre — egyedire vagy akár több lokális hálózatra is — installálni lehet.

Általában megállapítható, hogy a programok védelme elsősorban a forgalmazó érdeke (a speciális, forgalomba egyáltalán nem kerülő programoktól eltekintve). Az adatvédelmi rendszer, ahol az egyes programokat és adatokat csak meghatározott személyek használhatják, viszont már a felhasználó érdekeit szolgálja.

Bartha Attila: A NORTON...

„PC-s körökben” — és főleg rendszerprogramozók között — a Norton név fogalom. E könyvben a már hazánkban is elterjedt és közkedvelt Norton rendszer használatát ismertetjük. Részletesen kitérünk a Norton **Commander**, a Norton **Editor**, a Norton **Guide**, a Norton **Utilities** kezelésére, valamint a Norton **Integrator** alprogramjainak önálló illetve batch-file-ban történő alkalmazására.

A Norton **Commander** olyan rezidens program, amely kényelmesebbé teszi a DOS használatát. Ezen kívül lehetőséget nyújt saját menürendszerek kialakítására, lehetővé teszi a file-ok paraméter szerinti keresését, a tartalomjegyzékek struktúrájának megjelenítését...

A Norton **Editor** rendkívül intelligens szövegszerkesztő program: alkalmas több file egyidejű feldolgozására, a szöveg formázott ki nyomtatására.

A Norton **Guide** — ha nem is hagyományos — adatbáziskezelő program. Segítségével saját „HELP” menüt hozhatunk létre, melyben a definiált címszavakkal az általunk meghatározott információk érhetők el. Mivel ez is rezidens program, a funkciói bármikor — más program futása közben is — elérhetők.

A Norton **Utilities** segítségével a lemezek „lelki világában” mélyedhetünk el. Nagyon hasznos funkciója a törölt file-ok visszaállítása.

A Norton **Integrator** alprogramjai gyakorlatilag a DOS hiányosságait pótolják.

Ajánljuk ezt a könyvet — és a Norton rendszer használatát — minden kezdő IBM PC alkalmazónak, továbbá minden olyan haladó felhasználónak, akinek figyelmét eddig elkerülte.

Ára: 150.-Ft 96 oldal

Orosz Judit: IBM PC-HÁLÓZATOK
(LAN rendszerek: **Novell NetWare, PC Network, 3Com3+**)

A kiadvány célja, hogy az IBM személyi számítógépeken alkalmazott lokális hálózatok alapjaival ismertessen meg. Az első három fejezet a kezdőknek szól, betekintést nyújt a hálózatok alapvető tulajdonságaiba. Kitér felépítési szabványaikra, piaci viszonyaikra, valamint a hálózatok alapját képező operációs rendszerrel, az **MS-DOS**-szal való kapcsolataikra. A könyv második fele a haladók ismereteit foglalja kiválóan felépített rendszerbe, erős elméleti alapot szolgáltatva ezáltal. Foglalkozik hálózatok egymáshoz kapcsolásával; **NETBIOS**-szal. Az alapok ismertetését a különböző hálózati implementációk leírásával teszi teljessé ezt a precízen felépített munkát a Szerző.

Mindezek figyelembevételével ajánljuk ezt a kiadványt a számítógépes hálózatok iránt érdeklődő kezdő és haladó szakemberek figyelmébe.

Ára: 123.-Ft 52 oldal

Dr.Úry László: SYMPHONY

A sorozat célja a SYMPHONY programrendszer használatának ismertetése.

Az első rész a SYMPHONY installálásával, az elektronikus lappal, valamint a SYMPHONY makroival és programozásával foglalkozik.

A második kötet a SYMPHONY kezelésével kapcsolatos alaptéchnikák legfontosabb eseteit tárgyalja, különös tekintettel az ablaktechnikára. Számos, sokszor egyedi, de nagyon találó mintapéldával szemlélteti a könyv a legcélszerűbbnek tűnő megoldásokat. A könyv oda-vissza hivatkozásokat tartalmaz az első kötettel, így használatuk célszerűen csak együtt lehetséges.

I.kötet ára: 160.-Ft 132 oldal

II.kötet ára: 133.-Ft 90 oldal

A további kötetek előkészületben

**LEVÉLCÍM: LSI ATSZ
1300 BUDAPEST, Pf. 114**

Ára: 97,- Ft

EltGUARD a SW'88 díjnyertes terméke

Az EltGUARD védelmi rendszer

IBM PC XT/AT vagy kompatibilis számítógépen PC/MS-DOS operációs rendszer alatt futtatható programok (COM és .EXE fájlok) másolás és illetéktelen felhasználás elleni védelmére alkalmas.



Ára: 138.000.-Ft + ÁFA



EltSECRET

Bővebb felvilágosítással, bemutatóval állunk rendelkezésére:

adatvédelmi rendszer,
mely nemcsak az adatokhoz
és programokhoz való hozzáférést
szabályozza, hanem adattitkosításra
(kódolásra) is lehetőséget ad.



Számítástechnika
és Reklámszervező GMK.
1536 Budapest, Pf. 390.
Tel.: 550-962

EltSECRET rendszer jellemzői:

- 64 hierarchia-szint a fájlok védelmére
- 127 alkönyvtárszint
- 2500 jogosult felhasználó
- 3 különböző bonyolultságú kódolási eljárás
- együttműködés lokális hálózati rendszerekkel
- felhasználói menürendszerek kialakítása
- kiegészítő hardverrel fokozott biztonság

Egy telephely összes számítógépéhez elegendő egy példány
EltSECRET rendszert vásárolni.

Ára: 76.000.-Ft + ÁFA

Bővebb felvilágosítással, bemutatóval állunk rendelkezésére:

